AD-A069 297 DECISION SCIENCE INC SAN DIEGO CALIF F/G 5/9

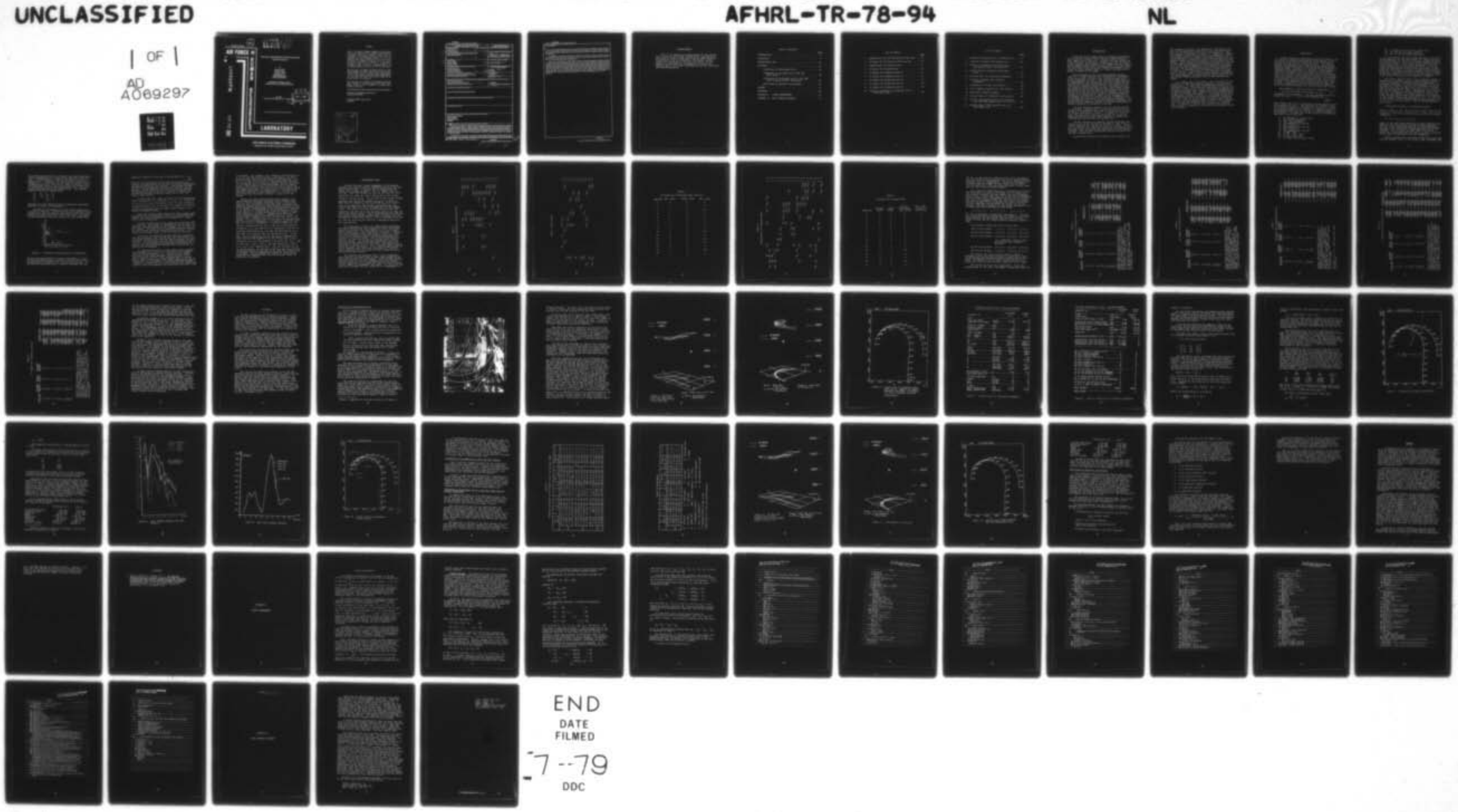
TACTICAL PERFORMANCE CHARACTERIZATION BASIC METHODOLOGY.(U)

MAY 79 M J WALSH, G H BURGIN, L J FOGEL F33615-78-C-0011

UNCLASSIFIED AFHRL-TR-78-94 NL

| OF |

AD
A069297



END
DATE
FILMED
7-79
DDC

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-

DDC QUALITY INSPECTED 1

LEVEL II

AFHRL-TR-78-94

AIR FORCE

HUMAN RESOURCES

AD A069297

DDC FILE COPY

TACTICAL PERFORMANCE CHARACTERIZATION: BASIC METHODOLOGY

By

Michael J. Walsh
George H. Burgin
Lawrence J. Fogel
Decision Science, Inc.
4901 Morena Boulevard
San Diego, California 92117

ADVANCED SYSTEMS DIVISION
Wright-Patterson Air Force Base, Ohio 45433

May 1979
Final Report for Period December 1977 – September 1978

Approved for public release; distribution unlimited.

LABORATORY

79 05 30 229

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE,TEXAS 78235

NOTICE

When U.S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This final report was submitted by Decision Science, Inc., 4901 Morena Boulevard, San Diego, California 92117, under contract F33615-78-C-0011, project 6114, with Advanced Systems Division, Air Force Human Resources Laboratory (AFSC), Wright-Patterson Air Force Base, Ohio 45433. Patricia Knoop (ASM) was the Contract Monitor for the Laboratory.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

GORDON A. ECKSTRAND, Technical Director
Advanced Systems Division

RONALD W. TERRY, Colonel, USAF
Commander

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFHRL-TR-78-94	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) TACTICAL PERFORMANCE CHARACTERIZATION: BASIC METHODOLOGY		5. TYPE OF REPORT & PERIOD COVERED Final rept. December 1977 – September 1978
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Michael J. Walsh George H. Burgin Lawrence J. Fogel		8. CONTRACT OR GRANT NUMBER(s) F33615-78-C-0011
9. PERFORMING ORGANIZATION NAME AND ADDRESS Decision Science, Inc. 4901 Morena Boulevard San Diego, California 92117		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62205F 61142306
11. CONTROLLING OFFICE NAME AND ADDRESS HQ Air Force Human Resources Laboratory (AFSC) Brooks Air Force Base, Texas 78235		12. REPORT DATE May 1979
		13. NUMBER OF PAGES 78
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Advanced Systems Division Air Force Human Resources Laboratory Wright-Patterson Air Force Base, Ohio 45433		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

adaptive maneuvering logic
air-to-air combat
linear programming
simulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This effort is to develop new methods of characterizing important features of tactical performance for display at an instructor/operator station of a flight simulator. In particular, the work included developing a technique for computing the weight or importance that a pilot assigns to various performance criteria. The work documented here represents the first of a two-phase program. Phase 1 involved developing the basic techniques and methods without collecting extensive pilot data. Phase 2 involves applying the methods to real pilot data collected on the Simulator for Air-to-Air Combat.

The approach was based upon a previously developed Adaptive Maneuvering Logic (AML) program. This program operates one-on-one against a real opponent to provide practice in combat flying. It operates by computing

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

387 228 LB

Item 20 Continued:

a "score" for each of several alternative next-moves and then executing the move rated highest. The score consists of a sum of weights assigned to each of the various criteria that would be satisfied if the move in question were chosen. The weights are fixed in the AML program. Thus, the program uses a fixed set of weights to produce a simulated performance.

The approach in this effort was to allow these weights to vary in fitting the output of the AML to that of a given pilot. In this way, the values of the weights may infer the importance that the pilot attaches to the various criteria and give valuable insight to his internal objectives.

Phase 1 work included using one AML program to emulate the pilot and developing the methods to compute the weights using a second AML program. It was found that for most criteria, the solution was fairly accurate and improved as more and more data were collected and used. Once the general mechanics were finished, the criteria themselves were examined with the goal of substituting new criteria that would be more useful to an instructor pilot. This effort revealed that to be of maximum utility, the criteria need to be maneuver-specific and should relate to the various flight maneuvers used in training combat tactics. Since the AML program was not designed to fly these specific maneuvers, work was directed toward modifying the AML accordingly. The report concludes with descriptions of these modifications and the successful use of the AML in flying a high speed yo-yo (a combat maneuver).

ACKNOWLEDGMENT

This is to express our appreciation for the personnel at Luke Air Force Base through whose cooperation we were able to obtain flight data from the SAAC simulator. In particular, we are indebted to Bob Bunker and Bob Coward for arranging for the data collection, to Bill Hopkins for the necessary programming to record the required data, and to the instructor pilots for performing the desired maneuvers.

TABLE OF CONTENTS

List of Tables

List of Figures

INTRODUCTION

Effective aerial combat requires adequate performance of the aircraft weapons system and full exploitation of this capability by the pilot. Combat training attempts to place the pilot in realistic situations so that he can gain a more complete understanding of the combat and can internalize behavior which would be most useful in actual combat. The instructor pilot scores the combat performance by the pilots as they simulate combat in flight simulators and in actual aircraft on the range.

Data displayed at the instructor/operator station (IOS) of a flight simulator must characterize performance sufficiently well to permit both instruction and proficiency assessment to occur. The problem of data portrayal is compounded by the requirement for succinctness due to a limited display area and the necessity for minimizing the instructor's workload imposed by the requirement to scan and integrate data from many sources. Data normally made available at the IOS are usually limited to status information about the aircraft and the environment. This type of data is plentiful, often requires considerable mental processing to meaningfully relate it to instructional requirements, and does not provide certain information that is fundamental to training and proficiency assessment. The effort reported here is to develop advanced techniques for characterizing important aspects of tactical performance in flight simulators for display at the IOS.

The more immediate objective of the current contract is to develop a method which by observation of pilot performance will determine the value or importance he assigns to various performance criteria. In Phase I of this effort, the task was to develop techniques for using the Adaptive Maneuvering Logic (AML) program to compute this information from recorded performance data.

The AML program is a computer program developed originally to act as an interactive opponent in real time on a flight simulator for one-on-one air-to-air combat. There is also a non-real-time (offline) version of the program using the same logic to simulate the maneuvering of two opposing aircraft. This offline program was used as the basis for the work reported here.

At each decision point (currently every second), the

AML pseudopilot projects the opponent's trajectory on the basis of the opponent's positions at the last three decision points and considers various trial maneuvers. These are elemental maneuvers and consist of segments of circular flight paths lying in a plane, called the maneuver plane. The flight path is specified by the rotation angle ρ of this maneuver plane, the throttle setting, and the applied load factor. Each maneuver is assigned a value equal to the sum of the weights corresponding to criteria which are satisfied by the relative geometry of the opponent's projected position and the projected position of the pseudopilot's aircraft. The maneuver with the highest value is chosen; in case of a tie, the maneuver plane closest to the opponent is chosen. Hence, the sum of the weights assigned to the chosen maneuver is always greater than or equal to the sum of the weights assigned to any of the rejected maneuvers.

It is seen, then, that given a set of weights for the criteria, the AML logic selects maneuvers on a second-by-second basis. In the effort reported here (Phase I of a two-phase study), techniques were developed to reverse this process in a sense, i.e., given a record of performance on a second-by-second basis, compute the set of weights which, if used by the AML, would allow it to perform identically. Further work planned for Phase II is to study actual pilot performance using this technique to determine by observation the values the pilot assigns to various performance criteria. Hence, a further task in Phase I was to analyze such criteria and determine their utility as a training aid. If possible, criteria with little or no training value should be replaced by criteria with increased utility for training.

DISCUSSION

The AML program was developed by Decision Science, Inc. to provide a computer program which would operate in conjunction with an aircraft simulator in an intelligently interactive mode and be a worthy opponent. The original program was developed under contract to the NASA Langley Research Center in support of the Differential Maneuvering Simulator (DMS) (Reference 1). Briefly, in the program, information relating to the situation is interpreted in terms of a valuated state space comprised of the relative values of acquiring various physical positions and orientations with respect to the opposing aircraft. The program then considers the alternative maneuvers for the aircraft it controls by examining the relative worth of the state entered. The maneuver with the highest state space value is then selected for execution and actions are taken to drive the simulated dynamics of the aircraft under control.

More exactly, a set of criteria or parameters x_1, x_2, . . ., x_n are considered with weights w_1, w_2, . . ., w_n assigned to the parameters. The value assigned to an instance $(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ is the weighted sum

$$\sum_{i=1}^{n} w_i \overline{x}_i.$$

The criteria are a set of questions with the answer to each being either yes or no. The questions are framed so that an affirmative answer is favorable and results in a value of 1 being assigned to the question. A negative answer is unfavorable and results in a value of 0 being assigned to the question. The questions in the version of the AML program used in the study are:

1. Is opponent in front of me?
2. Am I behind opponent?
3. Can I see opponent?
4. Can opponent not see me?
5. Can I fire 9L?
6. Can opponent not fire 9L?
7. Can I fire 9H?
8. Can opponent not fire 9H?
9. Is LOS $< 30°$?
10. Is $30° \leq$ LOS $< 60°$?
11. Is $60° \leq$ LOS $< 90°$?
12. Is range within sector limit?

13. Is range out of limit but improving?
14. Is rate of LOS within bounds?
15. Will I have an energy advantage?

At each second, the AML program considers several different maneuvers for the plane under its control, projecting them forward for from 3 to 8 seconds (projection time is an input value). The position of the opponent plane is extrapolated using its last three positions. Since the values assigned to the questions are either 0 or 1, the score for the maneuver is the sum of the weights of those questions which have a favorable response or equivalently a value of 1. The maneuver with the highest score is chosen by the AML program. In case of a tie, the maneuver whose flight plane is closest to the opponent is chosen.

The initial task in Phase I was to devise a method of determining the weights used by an AML program (pseudopilot) on the basis of observed choice of maneuvers. As indicated previously, for each trial maneuver, a score is assigned to it which is the sum of the weights for those questions which are assigned a value of 1; i.e., the answer to the question is "yes." The maneuver with the highest score is chosen by the AML pseudopilot. Hence, the score for this maneuver is greater than or equal to the score for each of the other trial maneuvers. For example, if the chosen maneuver had questions 1, 5, 7, 11, 12, and 15 with value 1 and a rejected maneuver had questions 1, 4, 6, 11, 12, and 14 with value 1, the following inequality would hold:

$$w_1+w_5+w_7+w_{11}+w_{12}+w_{15} \geq w_1+w_4+w_6+w_{11}+w_{12}+w_{14}$$

However, since w_1, w_{11}, and w_{12} occur on both sides of the inequality, they can be cancelled out, leaving the reduced inequality:

$$w_5 + w_7 + w_{15} \geq w_4 + w_6 + w_{14}$$

Hence, in the resulting inequality for each rejected trial maneuver, only the weights assigned to those questions peculiar to the chosen maneuver and to the rejected maneuver need to be considered. Common questions can be disregarded. Thus, at each decision time, a set of inequalities is generated and the task is to find a solution to the total set generated over the engagement.

Several schemes for solving the inequalities were considered; however, early in the study it was recognized that the problem was amenable to solution using the technique of linear programming, so it was used as the method of solution. (A description of linear programming is given in Appendix B.) In general, the set of values satisfying an inequality is a half-space in the space of weights, here, a 15-dimensional space. The set of values satisfying a set of inequalities is then the intersection of all the half-spaces corresponding to the inequalities. For example, consider the set of inequalities:

E1. $2w_1 + w_2 \leq 10$

E2. $w_1 + w_2 \leq 8$

E3. $w_2 \leq 7$

together with the standard linear programming requirement that the w_1 values are nonnegative.

A graphical representation of the three inequalities is given in Figure 1. The half-planes corresponding to the inequalities are indicated by the arrows so that the intersection or feasibility area is the lined area. Any point in the feasibility area will satisfy all the inequalities.

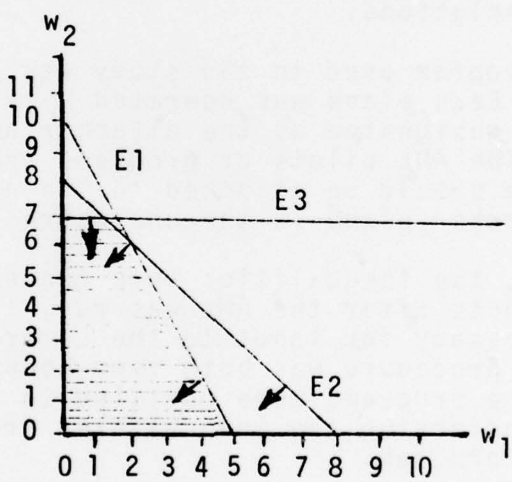


Figure 1. Graphical representation of inequalities.

In the usual application of linear programming, a linear function (termed objective function) is given which has to be maximized or minimized over the feasibility area. In the problem considered here, a natural candidate for the

objective function is the sum of the weights; i.e., $\sum_{i=1}^{n} w_i$.
With this as the objective function, for each set of weights, the program was run to obtain both the maximum and minimum solutions for the function. The maximum and minimum values give bounds on the possible solutions for the weights. In the different runs for most of the weights, the bounds were fairly tight; in several instances, the maximum and minimum values were equal and so give the actual value exactly.

In this study, the range of values for each weight was from 1 through 5; i.e., for each i, $1 \leq w_i \leq 5$. The possibility of $w_i = 0$ was eliminated, since if 0 was allowed as a possible weight, $w_i = 0$ for each i is always the minimum solution and so generally precludes tight bounds on the range of parameter values.

Since the actual weights used by the AML program under observation are a solution to the set of inequalities generated, the Linear Programming (LP) program will always find a set of solutions.

The AML program used in the study was an existing off-line version. Each plane was operated by its own AML pilot with one plane designated as the attacker and the other as the target. (The AML pilots or programs are equivalent so no significance should be attached to the names.) In this study, the attacker plane is the one which is observed.

Initially, the inequalities were manually extracted from the printouts after the AML was run, transformed into the format necessary for input to the LP program, and key-punched. This procedure was both time-consuming and prone to error, so the programs were modified to automate the process of transferring the inequalities from the AML program to the LP program.

As indicated previously in the report, each rejected trial maneuver gave rise to an inequality when compared with the chosen maneuver; i.e., the sum of the weights peculiar to the chosen maneuver (those weights in the chosen maneuver but not in the rejected one) is greater than or equal to the sum of the weights peculiar to the rejected one. In the program, for each considered maneuver, a number is formed with a 1 in each bit position (beginning right to left) corresponding to a question with value 1 and

0 elsewhere. For example, for a maneuver with questions 1, 5, 7, 11, 14, and 15 with value 1, the number (in octal) would be 62121 while for the maneuver with question 1, 4, 6, 11, 12, and 14 with value 1, the number would be 26051. The "exclusive or" of these two numbers (44170) would have 1's in exactly those bit positions where the two numbers differ. The "and" of this with each of the original numbers (62121 and 26051) would give rise to the numbers 40120 and 04050 which have 1's, respectively, in exactly those positions peculiar to each maneuver; i.e., in bits 5, 7, and 15 for the first and in bits 4, 6, and 12 for the second.

Hence, at each decision point (every second), the chosen maneuver with each rejected trial maneuver gives rise to a pair of numbers describing them. Successively applying the "exclusive or" and the "and" operations produces two numbers describing the questions peculiar to each. If the number corresponding to the rejected trial maneuver is 0, the pair is discarded as no new information is provided since it is already known that the weights are positive; otherwise, the pair is compared with the list of pairs already obtained from previous decision points. If the new pair is implied by a pair in the list, it is discarded. If it implies a pair in the list, it replaces that pair in the list; otherwise, it is added to the list, and the list count is incremented. A pair of numbers A_1, B_1 implies a second pair A_2, B_2, if the set of questions described by A_1 is the same as or is contained in the set of questions described by A_2, and the set described by B_1 is the same as or contains the set described by B_2. To see this, let $|A|$ denote the sum of the weights denoted by A. Then since all the weights in A_1 are in A_2, $|A_2| \geq |A_1|$. Similarly since all the weights in B_2 are in B_1, $|B_1| \geq |B_2|$ so that $|A_2| \geq |A_1| \geq |B_1| \geq |B_2|$ and hence $|A_2| \geq |B_2|$; i.e., $|A_1| \geq |B_1|$ implies $|A_2| \geq |B_2|$ and the pair A_2, B_2 can be discarded. At the end of the run, the list of pairs is printed out and is punched out on cards for input to the LP program. The LP program was recoded to accept these cards and to transform them into the internal format required by the program.

EXPERIMENTAL DATA

Initial runs were single engagements of 60 seconds duration with various initial conditions and with the data manually extracted. A typical set of inequalities for these runs is given in Table 1 with 26 inequalities resulting. Note that weights w_1, w_2, w_6, and w_8 occur only negatively; w_4 does not occur; and w_5 and w_7 are always paired as are w_6 and w_8. The results of the linear program together with the actual weights are given in Table 2. Note that the weights w_3 and w_9 through w_{15} show restriction on the bounds of the maximum and minimum and that those occur both positively and negatively in the inequalities. The sum $w_5 + w_7$ also occurs both positively and negatively and the sum of the maximums equals 5 as does the sum of the actual values. The LP program restricts the sum but cannot differentiate between them. Actually, any two nonnegative values which sum to 5 can be assigned to w_5 and w_7, and the resulting set would be a maximum solution to the inequalities.

As a test case, a set of 21 inequalities satisfying the set of weights was then prepared in which each weight occurred both positively and negatively and in at least three inequalities. The inequalities are given in Table 3, and the LP program results are shown in Table 4. Originally, the maximum was obtained without the minimum constraint of 1 on each weight. The resulting maximum (column four of Table 4) did not fit the data well, having three 0 results. The LP maximum program was then rerun with the minimum constraints included. The results are presented in column five of Table 4 and much better approximate the actual solutions. In all ensuing applications of the LP program, both the lower constraint of 1 and the upper constraint of 5 were used in both minimum and maximum solution derivations.

The inequalities obtained from a single engagement were found to be insufficient to give tight bounds on the possible solutions for the weights. Since the AML program has provisions for multiple engagements with different initial conditions in a single run, several runs with different numbers of engagements were made. For runs with the same set of weights for the target for all engagements in

Table 1

Inequalities for Preliminary Real Data Run

$$
\begin{aligned}
W_5 + W_7 + W_9 - W_{10} + W_{14} - W_{15} &\le 0 \\
W_{10} - W_{11} - W_{15} &\le 0 \\
-W_1 - W_{11} + W_{15} &\le 0 \\
-W_{10} + W_{11} &\le 0 \\
-W_{10} + W_{11} - W_{14} + W_{15} &\le 0 \\
W_9 - W_{10} + W_{14} &\le 0 \\
-W_9 + W_{10} - W_{13} - W_{14} &\le 0 \\
-W_5 - W_7 - W_9 + W_{10} - W_{14} &\le 0 \\
-W_5 - W_7 - \quad - W_{12} + W_{13} - W_{14} &\le 0 \\
W_{14} - W_{15} &\le 0 \\
W_9 - W_{11} + W_{14} - W_{15} &\le 0 \\
-W_3 + W_{12} - W_{13} + W_{15} &\le 0 \\
-W_{11} - W_{14} + W_{15} &\le 0 \\
-W_1 - W_{11} + W_{12} &\le 0
\end{aligned}
$$

Table 1 (Continued)

$$
\begin{aligned}
- W_5 - W_7 - W_9 + W_{10} &\leq 0 \\
- W_6 - W_8 + W_{15} &\leq 0 \\
W_3 + W_9 - W_{10} &\leq 0 \\
- W_3 - W_9 + W_{11} + W_{13} &\leq 0 \\
- W_9 + W_{11} - W_{14} &\leq 0 \\
W_3 - W_{13} &\leq 0 \\
- W_{13} + W_{14} &\leq 0 \\
- W_3 + W_{12} &\leq 0 \\
- W_3 + W_{11} &\leq 0 \\
- W_2 - W_9 + W_{10} - W_{14} &\leq 0 \\
W_3 - W_{15} &\leq 0 \\
- W_2 - W_3 + W_{15} &\leq 0
\end{aligned}
$$

Table 2

LP Output for Preliminary Real Data Run

Question	Min. Value	Actual Value	Max. Value
1	1	3	5
2	1	4	5
3	1	2	2.5
4	1	5	5
5	1	2	5
6	1	4	5
7	1	3	0
8	1	5	5
9	1	1	2.5
10	2	5	5
11	1	2	2.5
12	1	1	2.5
13	2	4	5
14	1	3	2.5
15	2	5	5

Table 3

Inequalities for Prepared Data

$$
\begin{aligned}
W_3 + W_5 + W_9 - W_{11} + W_{12} - W_{13} &\le 0 \\
W_1 + W_2 - W_3 - W_4 &\le 0 \\
- W_2 + W_3 &\le 0 \\
-W_1 - W_2 + W_4 &\le 0 \\
W_4 - W_5 - W_6 &\le 0 \\
- W_6 - W_7 + W_8 &\le 0 \\
W_9 - W_{10} + W_{11} + W_{12} &\le 0 \\
W_9 - W_{11} + W_{12} &\le 0 \\
W_{11} + W_{12} - W_{14} &\le 0 \\
W_7 - W_8 + W_9 &\le 0 \\
W_5 + W_7 - W_8 &\le 0 \\
- W_3 - W_4 + W_{11} + W_{12} + W_{13} &\le 0 \\
W_2 + W_5 - W_{10} - W_{12} &\le 0 \\
W_{12} + W_{13} - W_{15} &\le 0 \\
W_{12} - W_{13} - W_{14} + W_{15} &\le 0 \\
W_{10} - W_{13} + W_{14} - W_{15} &\le 0 \\
W_3 + W_5 - W_{10} &\le 0 \\
W_8 + W_{11} - W_{14} - W_{15} &\le 0 \\
-W_1 - W_2 + W_4 + W_5 &\le 0 \\
W_6 - W_8 + W_9 &\le 0 \\
W_{11} + W_{14} - W_{15} &\le 0
\end{aligned}
$$

Table 4

LP Output for Prepared Data

Question	Minimum Value	Actual Value	No Min. Constraints Max. Value	Max. Value with all Constraints
1	1	3	5	3
2	4	4	5	5
3	2	2	5	3
4	4	5	5	5
5	1	2	0	1
6	3	4	5	4
7	1	3	5	4
8	4	5	5	5
9	1	1	0	1
10	4	5	5	5
11	2	2	2.5	2
12	1	1	0	1
13	3	4	5	4
14	3	3	2.5	3
15	5	5	5	5

the run, no new data were obtained after three engagements (for any attacker being modeled, the set of weights must be invariant over all engagements). Runs were then made with different sets of weights for the target and different initial conditions. More data were obtained in these cases than for the invariant target weights.

As a result of these test runs, 5 production runs were made with each run having a different set of weights for the attacker but with the same weights for each engagement within a run. Within each run, the target had two different sets of weights with 3 engagements for each set of weights. Each set of 3 engagements had the same initial conditions: one where neither had an advantage, one where the attacker had the advantage, and one where the target had the advantage. One set of weights for the target consisted of all 1's while the other consisted of the values

1, 5, 3, 1, 4   4, 1, 1, 5, 3   1, 4, 1, 2, 2

for the 15 questions, respectively (see page 7). The test runs indicated that the second set of weights led to better performance by the AML program than did the first set. Hence, in each run the target presents different capabilities.

The attacker weights for the five runs were:

Run #1 with weights:	1,1,1,1,1	1,1,1,1,1	1,1,1,1,1
Run #2 with weights:	1,5,3,1,4	4,1,1,4,3	1,4,1,2,2
Run #3 with weights:	0,5,3,0,4	0,0,0,5,3	1,4,1,2,2
	i.e., questions 1,4,6,7, and 8 are deleted from attacker's decision.		
Run #4 with weights:	2,4,3,1,4	2,3,1,5,4	3,5,3,4,3
Run #5 with weights:	1,5,5,1,5	1,1,5,1,5	5,1,1,5,1

The results for these runs are given in Tables 5 through 9, which give actual weights of questions as well as the maximum and minimum solutions found by the LP program. The number pairs (in octal) representing the inequalities obtained from the AML program are also listed.

The results of Run #1 are predictable. Since the weights were all the same, the number of questions peculiar

Table 5

LP Output for Production Run #1

QUESTION	MAXIMUM WEIGHT	ACTUAL WEIGHT	MINIMUM WEIGHT
1	5	1	1
2	5	1	1
3	5	1	1
4	5	1	1
5	5	1	1
6	5	1	1
7	5	1	1
8	5	1	1
9	5	1	1
10	5	1	1
11	5	1	1
12	5	1	1
13	5	1	1
14	5	1	1
15	5	1	1

INEQUALITIES

44000	10004
4	40000
12005	44000
1000	2000
31001	44000
21000	2010
21000	4
2005	4010
21005	4010
1005	10010
1004	12000
4	10
22001	40014
21000	400
40004	2001
44014	12001
4000	10000
10004	4000
2	4
4002	10004
10010	4
4250	4
20520	42000
20520	41000
20521	40010
520	1010
520	2010
10016	44000
2005	2
14	10000
40004	2
10002	2001
21001	40000
10012	40000
22001	10002

Actual value and maximum and minimum values from LP program of weights satisfying the inequalities corresponding to listed pairs of octal numbers. A 1 in bit position i means weight w_i is included in the sum of weights for that number: sum of weights for that number $\geq$ sum of weights for second number.

Table 6

LP Output for Production Run #2

QUESTION	MAXIMUM WEIGHT	ACTUAL WEIGHT	MINIMUM WEIGHT
1	5	1	1
2	5	5	2
3	4	3	2
4	2	1	1
5	5	4	1
6	5	4	1
7	5	1	1
8	5	1	1
9	5	5	3
10	3	3	2
11	1	1	1
12	4	4	2
13	2	1	1
14	2	2	1
15	2	2	1

INEQUALITIES

400	41000		
20400	52010	5000	10400
44000	10004	520	42010
4	40000	30400	46000
2001	40000	45000	30400
1000	42000	30400	5010
21000	52000	30400	6010
400	11000	520	41010
20400	41010	240	40000
404	12010	2	40000
20000	10	2	4
40000	20000	10002	20004
61000	400	4	10010
400	42000	1004	22000
10404	5000	40400	21000
400	6000	402	21000
20000	40000	40000	10000
22001	50000	50010	4
51000	400	40000	10
1000	2010	40004	2
31001	44000	10010	40000
20400	5000		
4120	30000		

Actual value and maximum and minimum values from LP program of weights satisfying the inequalities corresponding to listed pairs of octal numbers. A one bit position i means weight w_i is included in the sum of weights for that number: sum of weights for first number $\geq$ sum of weights for second number.

Table 7

LP Output for Production Run #3

QUESTION	MAXIMUM WEIGHT	ACTUAL WEIGHT	MINIMUM WEIGHT
1	-	-	-
2	5	5	2
3	3.5	3	2
4	-	-	-
5	5	4	1
6	-	-	-
7	-	-	-
8	-	-	-
9	5	5	4
10	3	3	2
11	1	1	1
12	4	4	3
13	1	1	1
14	2.5	2	1
15	2	2	1

INEQUALITIES	
400	41000
20404	12000
400	42000
44,000	12004
4	50000
40000	2000
50000	20000
1000	42000
21000	52000
20000	40000
21000	2004
42004	21000
20400	5000
10004	4000
40004	1000
2	20000
22000	50000
50002	22000
2	4
4000	10000
400	11000
51000	400
10002	2000
50000	4
400	6000
4020	30000
5000	10400
30400	46000
45000	30400
11000	2004

Actual value and maximum values from LP program of weights satisfying the inequalities corresponding to listed pairs of octal numbers. A 1 bit position i means weight w_i is included in the sum of weights for that number: sum of weights for first number $\geq$ sum of weights for second number.

Question 1, 4, 6, 7, and 8 were not used in this run.

Table 8

LP Output for Production Run #4

QUESTION	MAXIMUM WEIGHT	ACTUAL WEIGHT	MINIMUM WEIGHT
1	5	2	2
2	5	4	2
3	5	3	2
4	3	1	1
5	5	4	1
6	1	2	1
7	5	3	1.5
8	3	1	1
9	5	5	2
10	4	4	1.5
11	3	3	1
12	5	5	2.5
13	4	3	2
14	5	4	2.5
15	4	3	2

Actual value and maximum and minimum values from LP program of weights satisfying the inequalities corresponding to listed pairs of octal numbers. A one bit position i means weight w_i is included in the sum of weights for that number: sum of weights for first number $\geq$ sum of weights for second number.

INEQUALITIES

402	41000	61004	12010
20400	12000	20004	10010
20400	42010	2	4
44000	10004	2	40000
4	40000	50000	4
2001	40000	20400	6000
1000	2000	6000	400
21001	50000	6000	11000
21000	12000	1001	11000
400	1000	404	1002
404	11000	2002	400
20400	1010	2002	1000
20000	10	401	10002
52000	21000	10004	4000
52010	20400	2001	14
41000	400	21010	42004
20000	40000	21001	40004
20400	41000	31001	44000
20400	5000	4000	10000
520	42010	401	40000
10400	6000	5000	10400
400	2000	520	41010
10404	5000	4120	30000
22001	50000	240	40000
42000	1000	10000	40000
21000	42000	2	10
60404	12010	40004	2
20410	52000	10002	2001
42004	11010	10010	4
40004	10010	40000	240
4	10000	10002	1001
40000	10	31000	400

Table 9

LP Output for Production Run #5

QUESTION	MAXIMUM WEIGHT	ACTUAL WEIGHT	MINIMUM WEIGHT
1	5	1	1
2	5	5	3
3	5	5	3
4	1.7	1	1
5	5	5	3
6	5	1	1
7	5	1	1
8	5	5	1
9	1.7	1	1
10	5	5	3
11	5	5	3
12	1.7	1	1
13	1.7	1	1
14	5	1	3
15	1.7	1	1

INEQUALITIES

402	41000
20404	12000
20400	42000
4	44000
4	50010
40000	10000
2001	40000
1000	2000
21000	52000
1000	400
30002	40004
31000	40404
42010	20400
42000	1000
41000	20400
20400	41000
11001	44000
12	40004
4	2
1001	40000
50002	4
2	40010
20000	10010
2001	10002
404	11000

1010	404
12001	44000
30010	40004
20000	40010
1010	40400
42000	400
404	2000
20400	5000
520	41010
520	42010
20400	6000
11000	6000
46000	11000
4120	30000
45000	10400
10000	4000
10000	40000
240	40000
22001	50002
12000	1000
40014	10002
10	40000
22001	40004
2005	10012
250	4
40004	12

Actual value and maximum and minimum values from LP program of weights satisfying the inequalities corresponding to listed pairs of octal numbers. A one in bit position i means weight w_i is included in the sum of weights for that number: sum of weights for first number $\geq$ sum of weights for second number.

to the chosen maneuver must exceed or be equal to the number of questions peculiar to the rejected ones. Hence, any set of equal weights would be a solution; since the solution of all 1's is the absolute minimum and the solution of all 5's is the absolute maximum for the objective function, these would be the minimum and maximum solutions.

In general, weights 2, 3, 4, and 9 through 15 are approximated and bounded fairly well. On the other hand, weights 1, 5, 6, 7, and 8 are not. On checking the questions, one sees that question 1 is not independent but is implied by questions 9, 10, and 11 and probably questions 5 and 7. This dependency appears to be reflected in that weight 1 occurred almost exclusively on the high side of the inequality. Only in Runs #1 and #4 did it occur on the low side of an inequality; and in Run #4, it had a bound other than the maximum. In Run #1, of course, it fared as well as any question.

Weights 5 and 7 always occurred paired on the same side of the inequality as did weights 6 and 8 so that the LP program can only determine the sum of the weights necessary to satisfy the inequality and cannot evaluate them individually. For example, in Run #4 weights 6 and 8 have a sum of 4 in the maximum solution so that any 2 weights which are greater than or equal to 1 and add to 4 will be a solution. The program, because of the order in which it handles these weights, assigned 3 to weight 8 and 1 to weight 6, not a good solution. The reverse would be a good solution. In Run #5 the sum of the weights for parameters 5 and 7 had to be at least 4. The LP program in the minimum solution assigned 3 to weight 5 and 1 to weight 7, giving a good solution. Again, the reverse would have been a solution but not a good one, since this would give weights to the questions which are inverse to the actual weights.

In recap, the LP program extracts as much information from the AML generated inequalities as possible. It gives minimum and maximum values for parameter weights and so gives a range of values for the parameter weights. Since any member of the set of solutions satisfies the inequalities, an AML pilot with any member of the set of solutions as weights would perform in the given engagements exactly as the original AML pilot. Naturally, if the number of engagements were increased by using additional targets with different weights and/or different initial conditions, more information would be available to the LP program and tighter bounds could be obtained.

CRITERIA

The AML program was not designed to simulate a human pilot but was designed to be a "worthy opponent." Hence, the criteria for decision making used in the AML program were not necessarily intended to agree with the criteria used by a human pilot. One of the tasks of the present study was to review and evaluate the criteria and, if they had little or no training value, to develop, if possible, other criteria which could be substituted to increase the utility for training.

In order to become familiar with the criteria used by human pilots, several discussions were held with pilots at Miramar Naval Air Station, and also a debriefing session of pilots from the Air Combat Maneuvering Range (ACMR) was attended. Analysis confirms that the relative geometry criteria (questions) used in the AML program are not the same as those used by pilots. Rather, they use the standard air combat maneuver appropriate to the situation.

One possibility considered for the AML program was the criteria which reflect pilot logic and/or training and which still allow the AML logic to fly the plane in a meaningful manner. Several sets of criteria were studied but could not be made to fit the short-term, look-ahead procedure of the AML logic under general flight conditions. The short-term AML maneuvers are done with a single command (the maneuver plane, the load factor, and throttle setting are specified), while in general, a sequence of commands is required to accomplish the more global maneuvers of the pilot.

This raised the question of whether or not the AML program using the relative geometry criteria could simulate the pilot flying the standard air combat maneuvers. A study of AML runs shows that under proper conditions the AML program does fly scissors and defensive turns. However, when confronted with situations that dictate a high-speed yo-yo, the AML does not fly the high speed yo-yo. Analysis indicates that in order to fly the yo-yo, the AML would require different weights over different parts of the flight and different trial maneuvers. It was then decided to look into modifying the AML program so that it would execute high-speed yo-yo's.

Simulation of High-Speed Yo-Yo

While trying to program the AML to execute high-speed yo-yo's, it became rapidly evident that--despite the fact that this maneuver has been instructed and used in air combat for years--it is still ill-defined and no analytical work defining and analyzing the high-speed yo-yo was found. The typical description of a high-speed yo-yo, as given in Tactical Manual NAVAIR 01-245 FDB-1T, Section I, Part 1, Figure 1-4, is as follows:

> When the overshoot appears imminent, the F-4 should roll a quarter turn away and pull up into the vertical plane ☆1.* This allows nose-tail separation to be maintained. Afterburner may be employed as required to maintain closure.
>
> After starting the pull-up, the F-4 should keep the nose coming up and roll toward the enemy to keep him in sight. At the slower speed in the apex ☆2, the F-4 should pull his nose back down through the horizon to realign with the enemy's six o'clock position ☆3.

The maneuver is illustrated in Figure 2. It is, of course, almost impossible to translate such statements as, "When the overshoot appears imminent, . . ." into a computer program without some method of translating all these qualitative statements into quantitative statements.

The most efficient way to obtain quantitative data appeared to be to record the performance of a high-speed yo-yo by an experienced instructor pilot on a simulator and to use these data as a baseline for modifying the AML program so that it can perform high-speed yo-yo's. In addition to the performance of a "perfect" high-speed yo-yo, it was planned to have a few high-speed yo-yo's with typical errors flown in order to get preliminary insight into types of errors to be encountered in Phase II.

This data collection was accomplished at Luke Air Force Base on the Simulator for Air-to-Air Combat (SAAC), where on 11 July 1978 an instructor pilot flew a series of eight high-speed yo-yo's against a noninteractive target in a defensive turn, some medium to good (in his own judgment), some purposely not so good. Time histories of these flights, consisting of position and attitude and their derivatives, were recorded on magnetic tape at one-half

---

*Numbers in ☆ refer to aircraft positions in Figure 2.

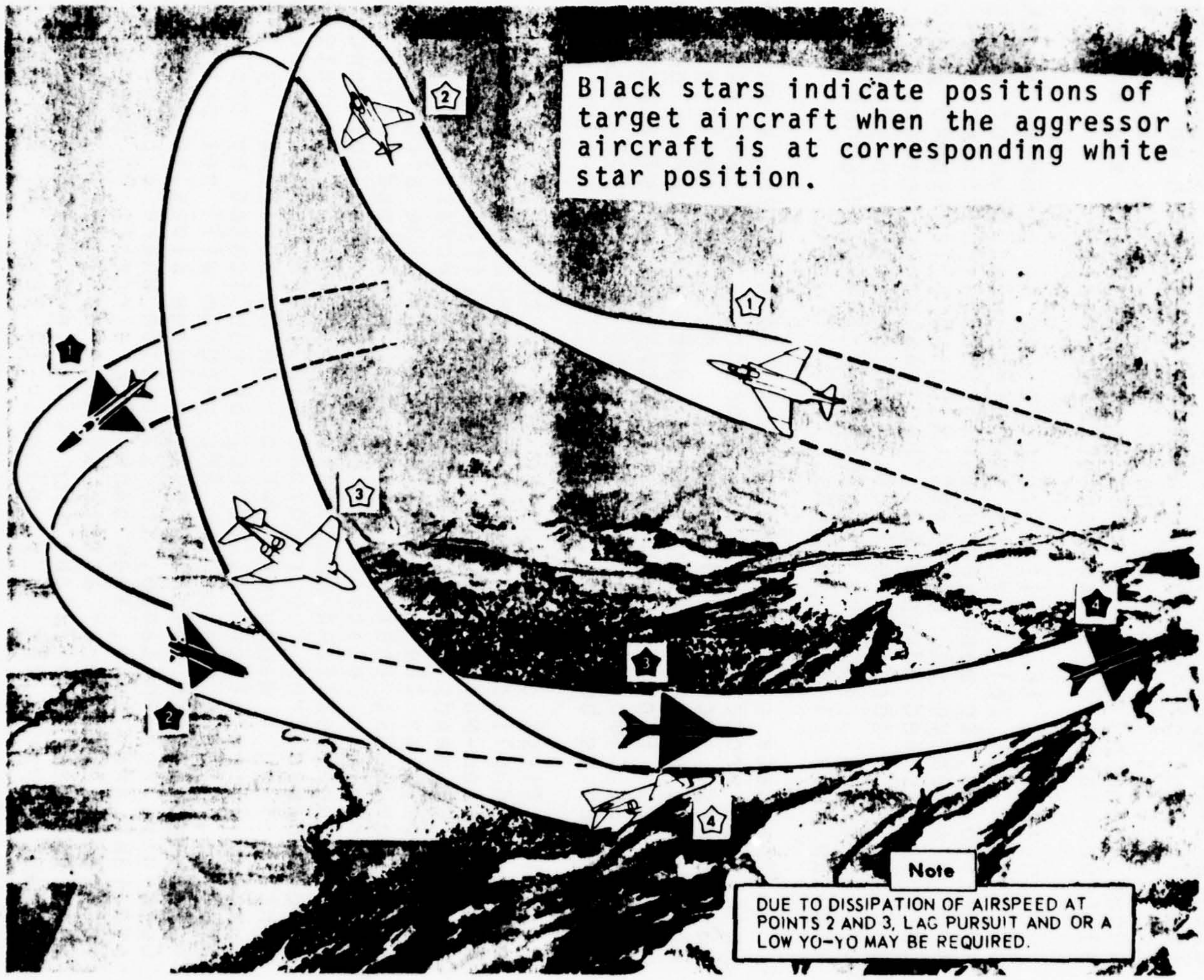


Figure 2. Classical textbook high-speed yo-yo.

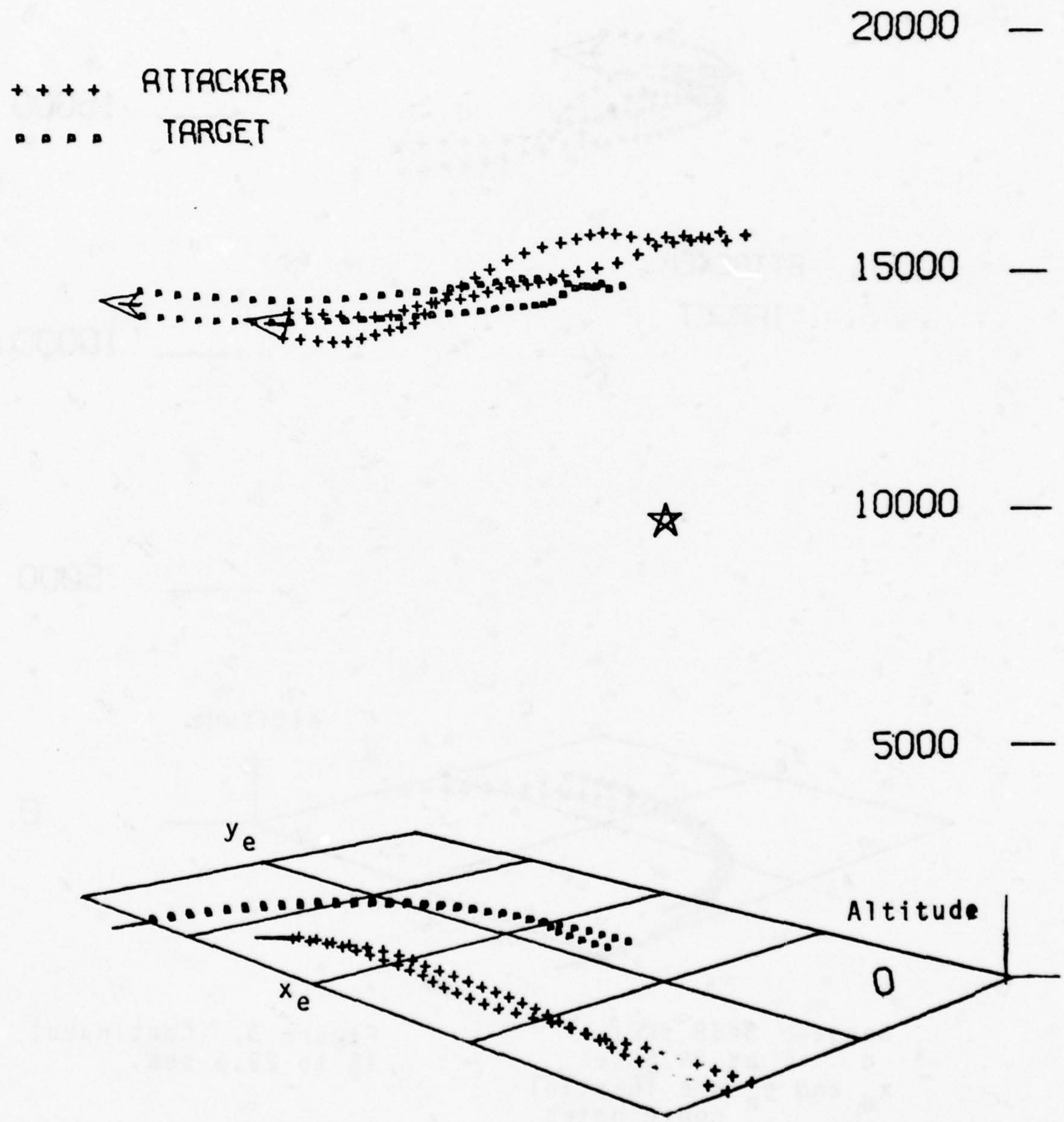
second intervals. This data base, recorded in 32-bit words in Sigma 5 format, was then converted to 48-bit words compatible with the AML program on the CDC 3600.

The first step of the analysis then established a reference high speed yo-yo. Figure 3 shows a "three-dimensional" plot of the run #2 at Luke AFB, which was judged by the pilot as the best of the yo-yo's he flew. Figure 4 shows a ground trace of this engagement with the aircraft altitude labelled at two-second intervals.

Note that the initial conditions as selected by the pilot did not call for the immediate execution of a high-speed yo-yo; to maneuver himself into a position requiring a high-speed yo-yo, he first executed a low-speed yo-yo to gain some speed advantage. The trajectory between $t = 0$ and $t = 11.5$ seconds reflects the low speed yo-yo portion of the flight, and the remainder is the high-speed yo-yo, with an apex at 22.5 seconds.

The data of the encounter as flown on the simulator were then processed by the AML program to obtain additional parameters, such as line-of-sight angle and angle off-tail, which were computed from the raw data as recorded on the simulator. Figures 5 and 6 illustrate the reference encounter at time 28 seconds, when the high-speed yo-yo was considered to be completed.

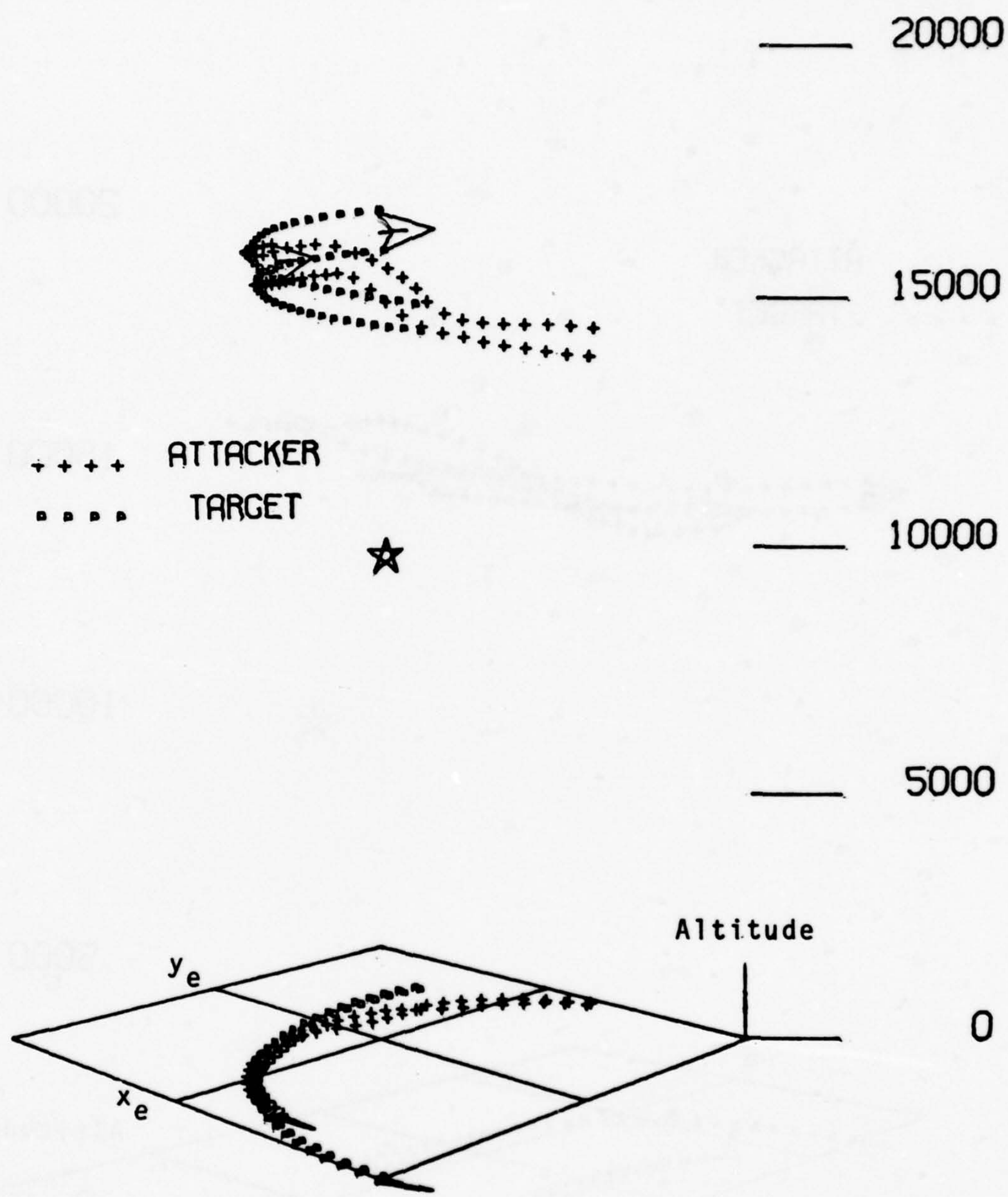
The first concern was to see if the reference run could be duplicated with the AML program by bypassing the AML decision-making routine and by specifying, at every second, commands to the AML attacker aircraft but in the same format as the standard AML defines the aircraft maneuver commands; that is, by specifying a load factor and a maneuver plane rotation angle (variable ρ in Reference 1). It was soon realized that the angle ρ is not accurately determinable from the recorded simulator raw data because certain flight maneuvers, which can be performed by the human-piloted aircraft, cannot be replicated by the AML program. For example, the AML program will not perform flight maneuvers which result in large sideslip angles. Consider, for instance, the situation where a pilot flies straight and level, then banks the aircraft 90 degrees and reduces the angle of attack so that no lift is generated. This results in a flight path lying in a vertical plane, concave towards the $x_e\ y_e$ plane. In terms of the AML program, this is a maneuver plane with a rotation angle of 180 degrees. To fly in such a plane, the AML aircraft will roll 180 degrees and then reduce the angle of attack to obtain zero lift. This results in a flight maneuver which



Range = 3956 feet at 14.5 sec.

Figure 3. 3D plot of reference high-speed yo-yo and ground trace --1 to 14.5 sec.

x_e and y_e are inertial coordinates



Range = 3458 feet
at 29.5 sec.
x_e and y_e are inertial coordinates

Figure 3. (Continued)
15 to 29.5 sec.

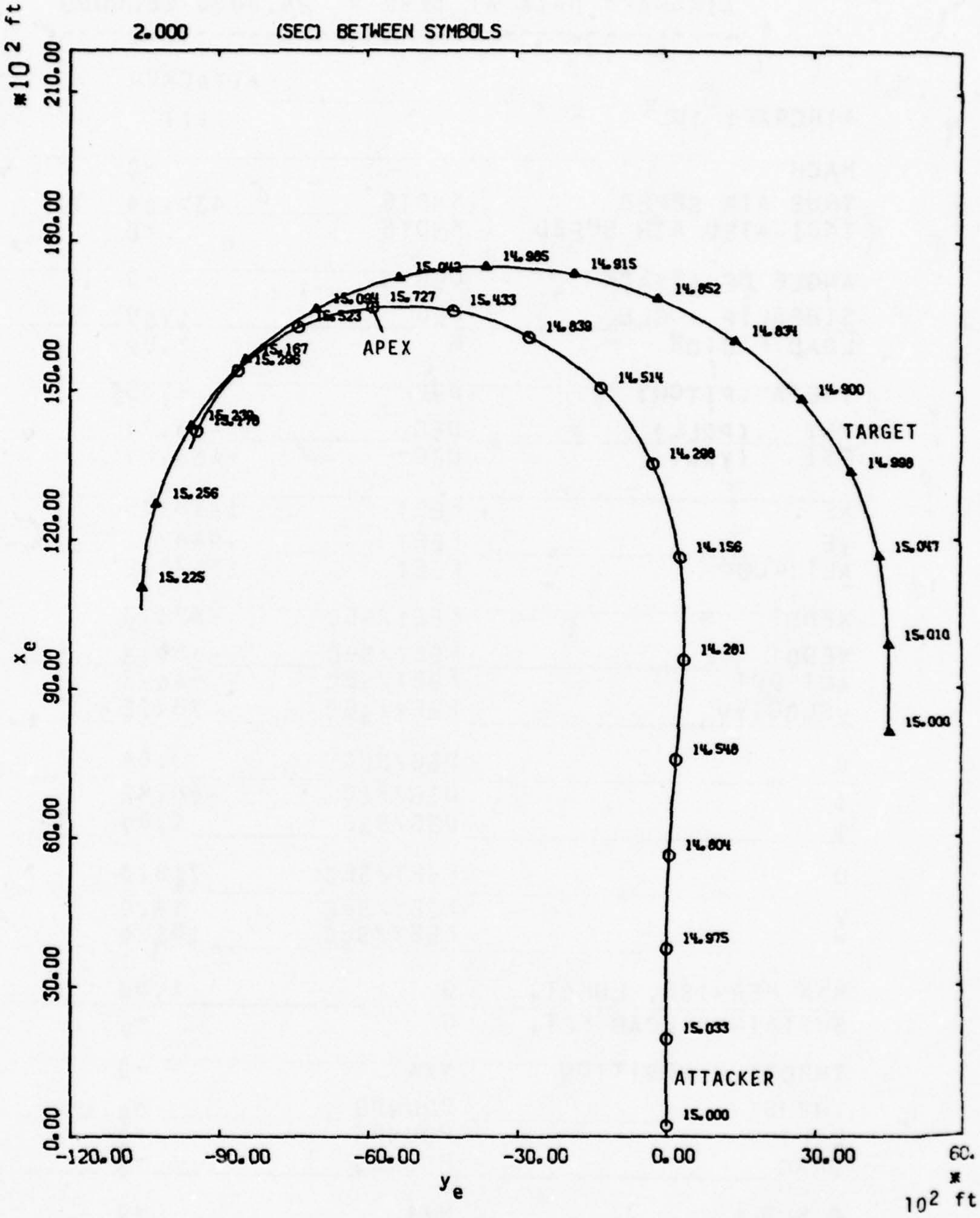


Figure 4. Ground trace of reference high-speed yo-yo. Numbers by symbols represent altitudes in kilofeet. (x_e and y_e represent inertial coordinates.)

AIRCRAFT DATA AT TIME = 28.0000 SECONDS

		ATTACKER	TARGET
AIRCRAFT ID.		;;;;	;;;;
MACH		-0	-0
TRUE AIR SPEED	KNOTS	435.04	507.14
INDICATED AIR SPEED	KNOTS	-0	-0
ANGLE OF ATTACK	DEG	-0	-0
SIDESLIP ANGLE	DEG	1.09	-.10
LOAD FACTOR	G	3.52	3.59
THETA (PITCH)	DEG	-.35	.63
PHI (ROLL)	DEG	-68.77	-75.16
PSI (YAW)	DEG	-168.27	173.38
XE	FEET	14181.5	11030.0
YE	FEET	-9447.1	-10561.1
ALTITUDE	FEET	15178.2	15224.6
XEDOT	FEET/SEC	-673.3	-855.3
YEDOT	FEET/SEC	-289.3	-22.6
ALT DOT	FEET/SEC	-46.2	-24.3
VELOCITY	FEET/SEC	734.2	855.9
θ	DEG/SEC	.04	.62
ϕ	DEG/SEC	-68.72	-75.02
ψ	DEG/SEC	1.09	-.10
U	FEET/SEC	718.3	846.7
V	FEET/SEC	14.0	-1.5
W	FEET/SEC	151.6	125.7
MAX PERMISS. LDFCT.	G	1.00	1.00
SUSTAINED LOAD FCT.	G	-0	-0
THROTTLE POSITION	N/A	-0	-0
THRUST	POUNDS	-0	-0
LIFT	POUNDS	-0	-0
DRAG	POUNDS	-0	-0
C SUB L	N/A	-0	-0
SPEC. ENERGY/1000	FEET	23.56	26.61
SPEC. ENERGY RATE	FEET/SEC	0	0

Figure 5. Aircraft data for reference engagement.

TACTICAL SITUATION AT TIME = 28.0000 SECONDS

		ATTACKER	TARGET
AIRCRAFT ID.		F-4	F-4
RANGE	FEET		3342.9
RANGE RATE	FEET/SEC		83.0
LINE OF SIGHT ANGLE (LOS)	DEG	7.82	153.91
AZIMUTH OF LOS (IN BODY AXES)	DEG	3.88	-172.61
ELEVATION OF LOS (IN BODY AXES)	DEG	-6.79	25.09
DEVIATION ANGLE	DEG	5.80	161.88
DEVIATION ANGLE RATE	DEG/SEC	-47.66	23.04
ANGLE OFF	DEG	26.09	172.18
ACCUMULATED OFFENSIVE TIME	SEC	19.0000	0
ACCUMULATED TIME FOR WEAPON 1	SEC	11.3600	0
ACCUMULATED TIME FOR WEAPON 2	SEC	11.3600	0
ACCUMULATED TIME FOR WEAPON 3	SEC	4.0000	0

01 IS OPPONENT IN FRONT OF ME	1	0
02 AM I BEHIND OPPONENT	1	0
03 CAN I SEE OPPONENT	1	0
04 CAN OPPONENT NOT SEE ME	1	0
05 CAN I FIRE 9L	0	0
06 CAN OPPONENT NOT FIRE 9L	0	0
07 CAN I FIRE 9H	0	0
08 CAN OPPONENT NOT FIRE 9H	0	0
09 IS LOS LESS THAN 30 DEGREES	6	0
10 IS LOS BETWEEN 30 AND 60 DEGREES	0	0
11 IS LOS BETWEEN 60 AND 90 DEGREES	0	0
12 IS RANGE WITHIN SECTOR LIMITS	1	0
13 IS RANGE OUT OF LIMITS BUT IMPROVING	0	1
14 RATE OF LOS WITHIN LIMITS	0	0
15 WILL I HAVE AN ENERGY ADVANTAGE	2	2
CELL NUMBER	18943	20480
CELL VALUE	13	3

Figure 6. Tactical situation for reference engagement.

creates no sideslip.

By specifying load factors and maneuver plane rotation angles, it was not possible to obtain the same bank angles that were present in the simulator flight. This situation is particularly pronounced for flight with low load factors because the gravity vector becomes relatively more important than the lift vector.

To be able to replicate the maneuvers flown on the simulator, the AML program was modified to accept as maneuver command the Euler roll angle and load factor instead of maneuver plane rotation angle and load factor.

The Euler roll angle ϕ can be calculated directly from the recorded direction cosine matrix C as

$$\phi = \arc\tan\,(c_{23}/c_{33}) \text{ where}$$

$$C = \left\{ \begin{matrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{matrix} \right\}$$

The load factor can be calculated from the given acceleration along the aircraft z-axis and from the aircraft attitude. For first approximation, a negligible angle-of-attack is assumed and thrust is aligned with the aircraft x-axis. Then, the force vector along the aircraft z-axis is equal to minus the lift plus the projection of the gravity force onto the aircraft z-axis. The projection of the gravity force is equal to

$$\text{Weight} \cdot \cos\theta \cdot \cos\phi$$

Also, the force along the aircraft z-axis is equal to $a_z \cdot$ Weight where a_z is the acceleration along the aircraft z-axis. Equating the two formulas for the force along the z-axis yields

$$a_z \cdot \text{Weight} = -\text{Lift} + \text{Weight} \cdot \cos\theta \cdot \cos\phi$$

Hence, the acceleration a_z is given by

$$a_z = -\frac{\text{Lift}}{\text{Weight}} + \cos\theta \cos\phi$$

Since, by definition, the load factor is equal to the lift/weight,

$$a_z = - \text{load factor} + \cos\theta \cos\phi$$

The third variable used to control an aircraft in air combat is thrust. The throttle setting during the runs on the Luke simulator was not recorded. It was set to afterburner in the simulations discussed here, and during the entire maneuver, the pilot apparently had his aircraft in afterburner, too.

The high-speed yo-yo was first simulated using the same roll angles and load factors as were recorded at Luke AFB. With the AML program, this resulted in a turn considerably too tight; also, terminal velocity was lower than that of the reference yo-yo. This may be caused by a different value of the drag between the simulator F-4 model and the AML F-4 model. For the purpose of our study, little benefit would be gained in trying to match the performance of the two models.

While operating the F-4 aircraft in the AML program with load factors as calculated by the above formula, it was observed that the AML program-driven aircraft decelerated faster than the aircraft as simulated on the SAAC. It was, therefore, decided to calculate the aircraft acceleration from the given flight path in order to validate the normal acceleration obtained from the recorded data from the SAAC. The positions of the aircraft at 17, 18, and 19 seconds were used to determine a circle which is probably a very good approximation of the actual flight path during these 2 seconds. The coordinates of the attacker aircraft at these three times were:

t	x_e feet	y_e feet	h feet	v feet/sec
17"	15,628	-2,002	14,645	914
18"	16,079	-2,756	14,839	907
19"	16,403	-3,516	15,117	897

The center of the circle determined by these three points lies at x_e = 12,063, y_e = -3,798, h = 18,036; the radius of this circle is 5,238 feet (see Figure 7).

The normal acceleration to that flight path.

$$a_n = \frac{V^2}{R} = 157 \text{ ft/sec}^2$$

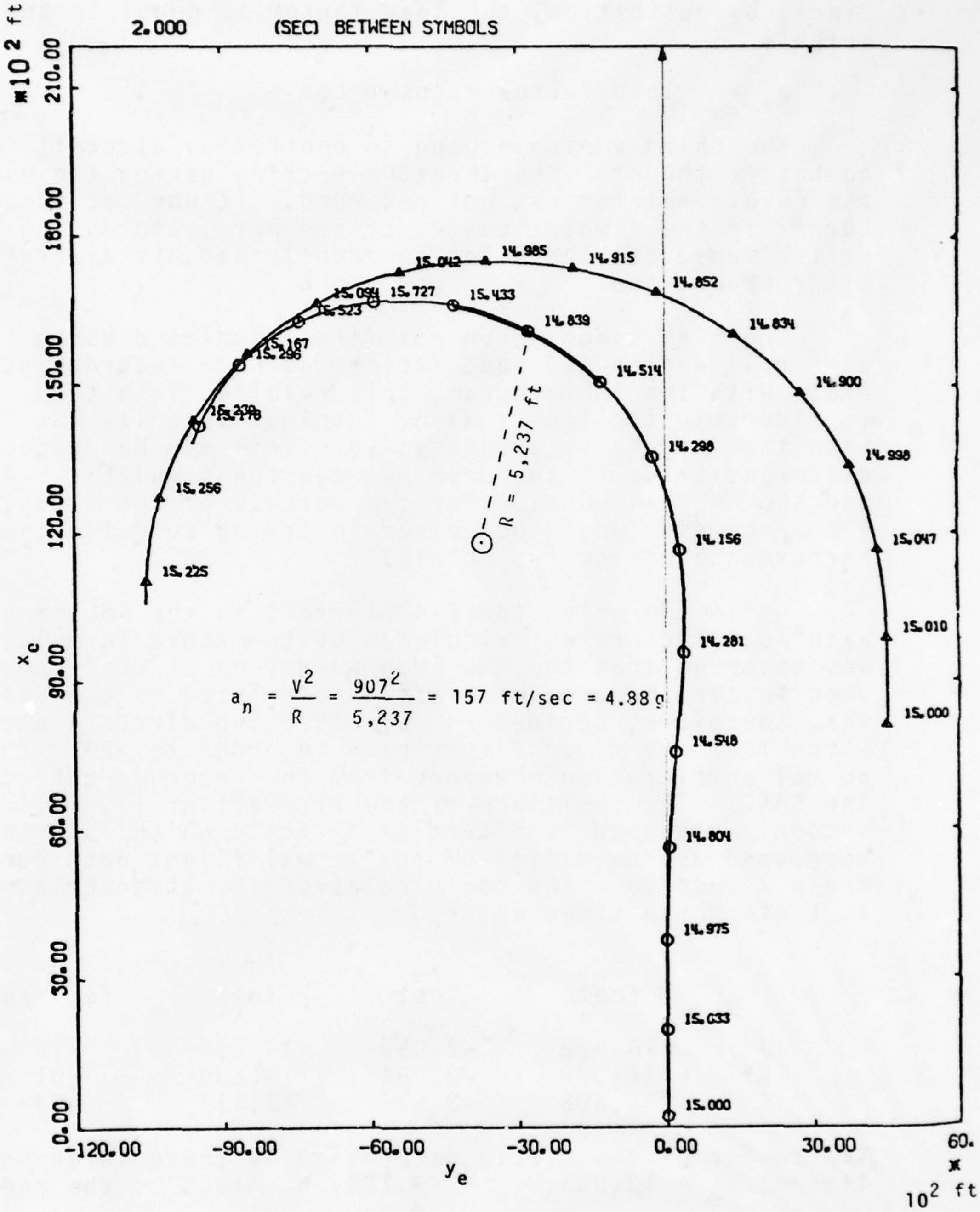


Figure 7. Estimation of normal acceleration.

$a_n = 4.88g$

The tangential acceleration is approximately 8 ft/sec^2 = 0.25g.

The total acceleration acting on the center of gravity of the aircraft is, therefore, less than 5g. The recorded accelerations along the aircraft z-axis at these three times were:

t sec	a_z g
17	6.55
18	5.99
19	6.06

It seems justified, therefore, not to use the recorded acceleration along the aircraft z-axis as basis for calculating load factors to be used by the AML program.

Recognizing the fact that the recorded normal acceleration might be too high, a trial command sequence for load factors as shown in Figure 8 was selected. After running cases 1, 2, and 3, it became obvious that an almost perfect high-speed yo-yo should be obtainable by adjusting load factors and bank angles only after the apex of the yo-yo. Case 6 on Figure 8 shows a command sequence of a good high-speed yo-yo. Figure 9 shows the corresponding command sequences for the bank angle; Figure 10 shows the ground traces of the different high-speed yo-yo's.

It is interesting to compare some of the pertinent terminal conditions between the reference yo-yo and case 6 (all data at 28 seconds):

	Reference Yo-Yo	Case 6
Line-of-Sight Angle	7.82 deg.	4.20 deg.
Deviation Angle	5.80 deg.	14.97 deg.
Angle-Off	26.09 deg.	16.12 deg.
Range	3,342 ft	2,937 ft
Range Rate	83 ft/sec	77 ft/sec
Velocity	734 ft/sec	794 ft/sec
Altitude	15,178 ft	15,057 ft
Specific Energy*	23,551 ft	24,870 ft

---

*Specific Energy is the sum of potential and kinetic energies divided by the weight.

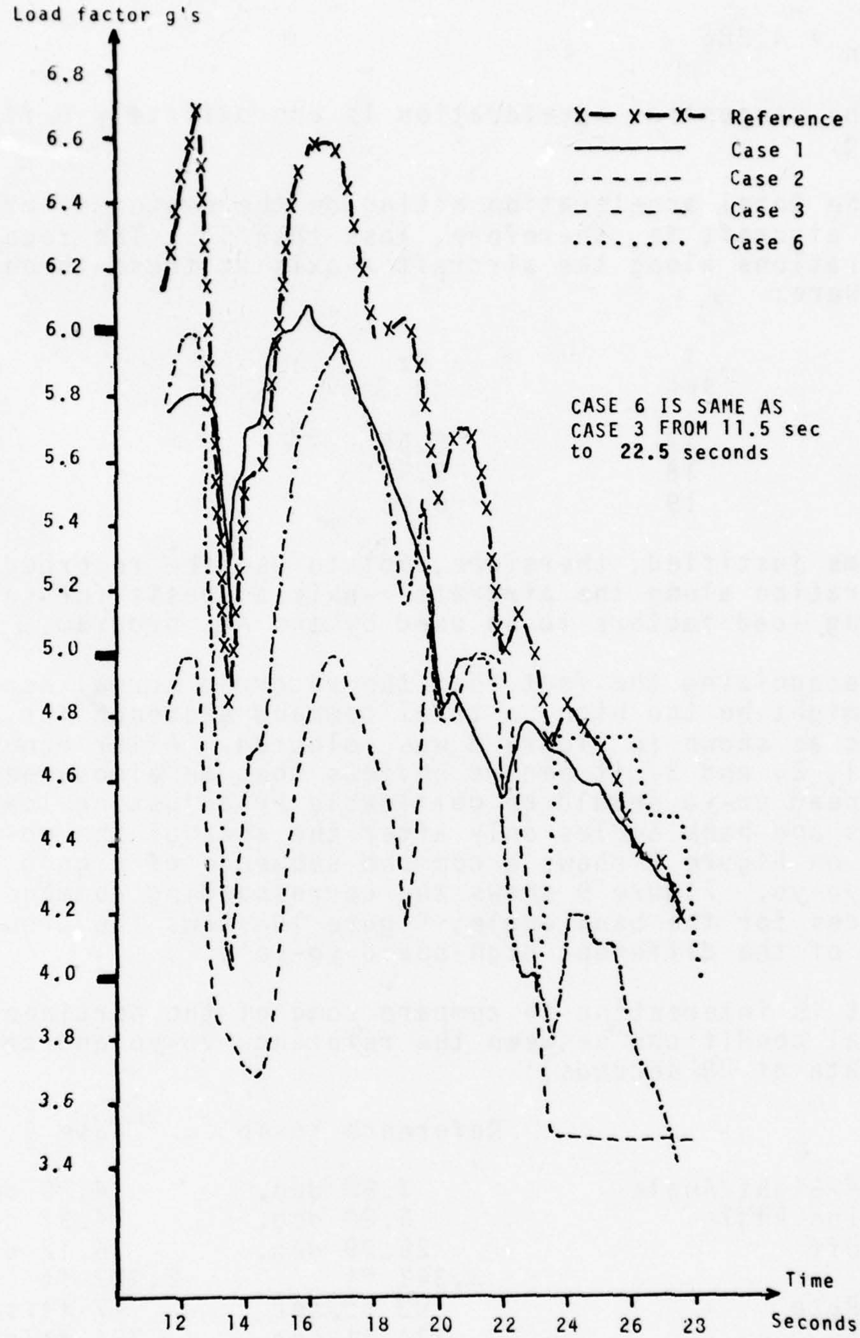


Figure 8. Trial command sequence for load factors.

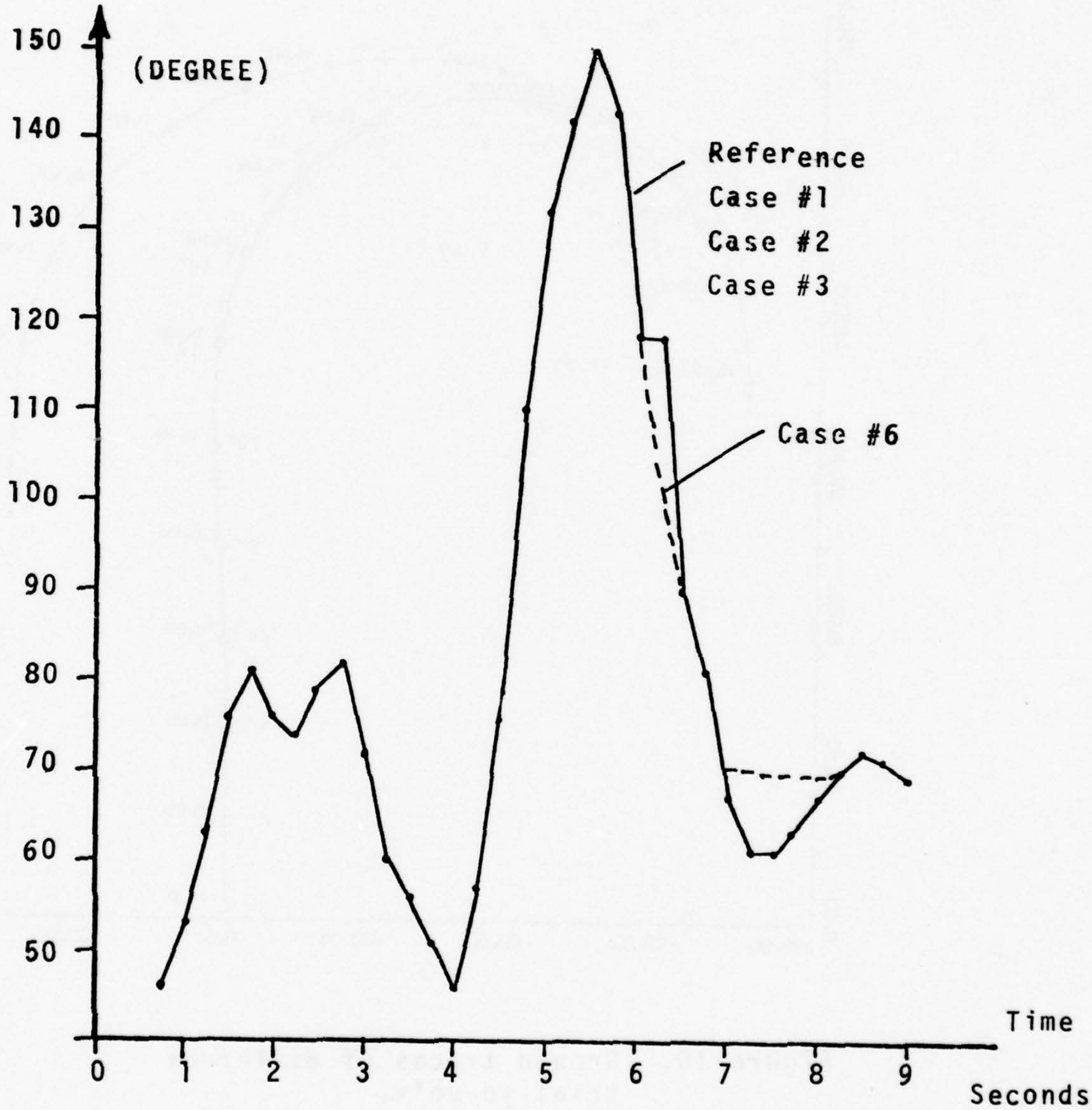


Figure 9. Bank angle command sequence.

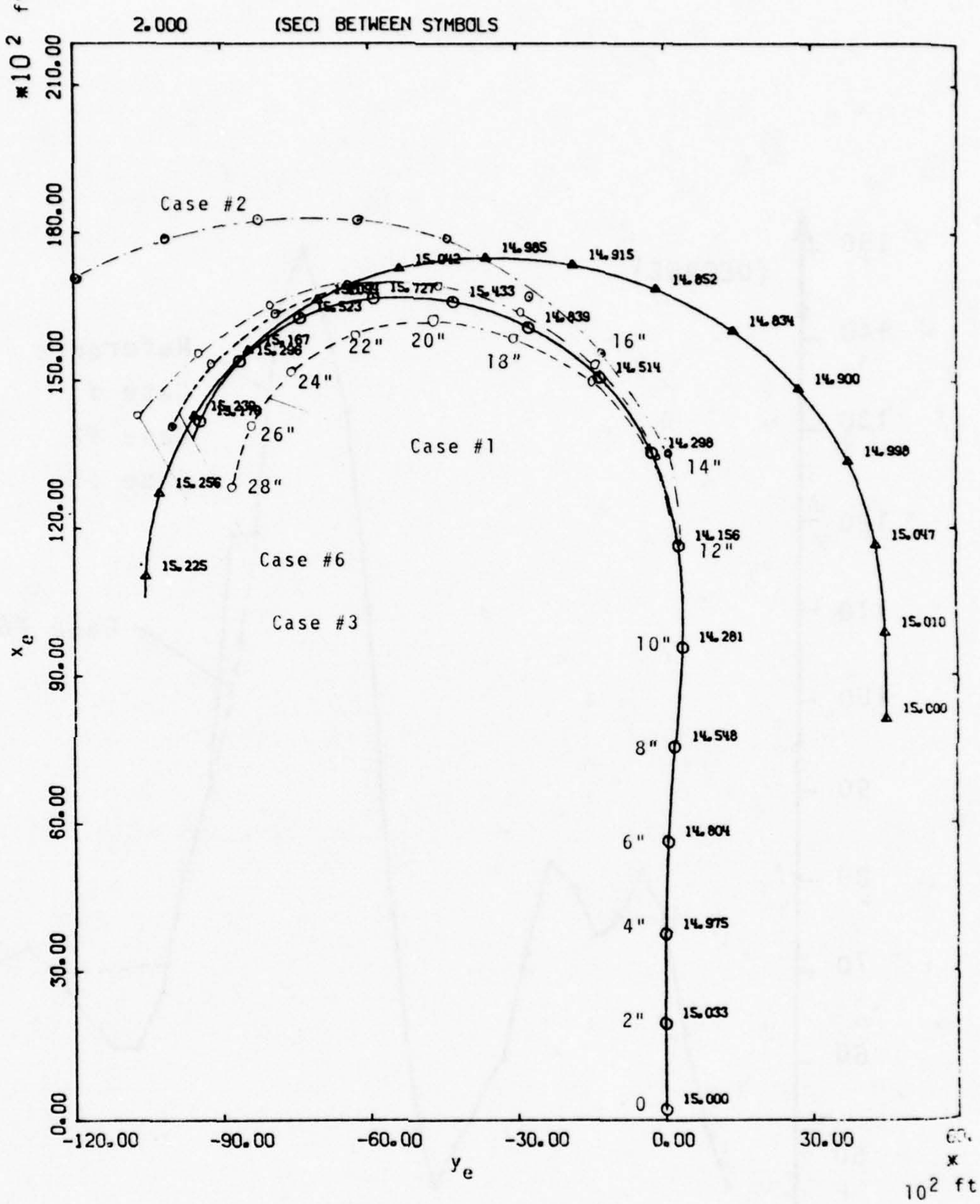


Figure 10. Ground traces of different trial yo-yo's.

It would appear that the yo-yo of case 6 is superior to the reference yo-yo for 3 reasons. Most important, the pilot ends up outside the turn of the defender, which, according to pilots from the Navy Fighter Weapons School at NAS Miramar, is desirable. The 60 ft/sec higher terminal velocity is certainly an asset, especially if the defender should try for another attack; and finally, the 10-degree difference in the angle-off gives him an advantage. Table 10 lists the values of some of the physical variables at various times in the different cases.

Simulation of Low-Speed Yo-Yo with the AML Program

Once a suitable command sequence for the high-speed yo-yo was found, the entire run 2 of the Luke simulator was "flown" by the AML program. Finding a suitable command sequence for a low-speed yo-yo is much simpler than for a high-speed yo-yo because g levels applied during a low-speed yo-yo are low during the entire maneuver.

Figure 11 shows the three-dimensional representation of the combined low- and high-speed yo-yo's, and Figure 12 shows the ground trace. Comparisons between Figure 11 and Figure 3 and between Figure 12 and Figure 4 show an almost identical execution of the low-speed portion of the flight while the AML-executed high-speed yo-yo appears to be somewhat better than the reference high-speed yo-yo.

Simulation of High-Speed Yo-Yo's with AML under Varying Initial Conditions

To demonstrate that not only high-speed yo-yo's for the same initial conditions as used in the reference yo-yo can be simulated by the AML program, the initial velocity of the attacker aircraft was increased from 1,037 ft/sec to 1,100 ft/sec and 1,150 ft/sec, cases 7 and 8.

The line-of-sight angle is the angle between the attacker aircraft's x-body axis and the line-of sight vector to the target aircraft. The deviation angle is defined as the angle between the attacker's velocity vector and the line-of-sight vector from the attacker to the target. Note that if sideslip angle and angle-of-attack were zero, the deviation angle would be the same as the line-of-sight angle.

The angle-off is defined as the angle between the line-line-of-sight vector from the attacker to the target and the target's velocity vector. Below are the terminal conditions for the 1,100 ft/sec initial velocity (at time 27.5 s):

Table 10

Physical Variables for Different Yo-yo's at Various Times

		V ft/s	Θ °	H kft	$\dot{H}$ ft/s	λ °	$\dot{\lambda}$ °/s	AOT °	R kft	$\dot{R}$ ft/s	SpEn. kft	1 2 3 4 *
	Ref	914	19.1	14.64	196	18.5	-4.02	49.4	3.10	-265	27.41	1 1 1 0
	#1	929	17.7	14.80	177	24.8	5.16	53.0	2.87	-302	28.21	1 1 1 0
	#2	1022	11.4	14.64	120	11.7	10.4	40.1	2.88	-237	30.88	1 1 1 0
17"	#3	980	15.0	14.7	152	10.4	4.65	46.3	2.83	-300	29.63	1 1 1 0
	#6	970	15.85	14.76	163	13.9	5.67	47.98	2.83	-307	29.39	1 1 1 0
	Ref	804	-14.55	15.73	-74	5.4	-1.96	33.5	2.82	18.6	25.8	1 1 1 0
	#1	774	-9.99	15.83	-47.9	23.3	5.56	48.3	2.35	-79	25.15	1 1 1 0
	#2	979	-11.2	15.32	-127	53.5	3.86	7.6	3.49	385	30.22	1 1 1 1
22"	#3	866	-9.55	15.65	-68	15.4	1.82	25.0	2.53	54	27.31	1 1 1 1
	#6	848	-9.4	15.75	-60	9.9	.56	29.8	2.46	18	29.92	1 1 1 1
	Ref	763	1.7	15.39	-110	4.3	.29	23.6	3.05	96	24.4	1 1 1 1
	#1	714	.86	15.49	-95	36.5	5.65	43.6	2.29	35.4	23.4	1 1 0 0
	#2	1012	-8.9	14.72	-208	65.99	2.83	7.9	4.86	513	30.64	1 1 1 1
25"	#3	871	-3.8	15.23	-133	24.3	1.82	9.3	2.85	139	27.04	1 1 1 1
	#6	830	-4.6	15.36	-133	10.25	-1.55	14.8	2.66	94	26.07	1 1 1 1

Table 10 (Continued)

	V ft/s	Θ °	H kft	$\dot{H}$ ft/s	λ °	$\dot{\lambda}$ °/s	AOT °	R kft	$\dot{R}$ ft/s	SpEn. kft	1 2 3 4 *
Ref	734	-.35	15.18	-46	7.8	-.79	26.1	3.34	83	23.55	1 1 1 1
#1	642	4.97	15.35	-15.1	58.08	2.62	51.4	2.555	140	21.6	1 1 0 0
#2	1048	-7.7	14.17	-176	71.2	.58	10.9	6.50	571	31.24	1 1 1 1
28" #3	886	-3.0	14.94	-83	31.5	2.25	6.8	3.297	151	27.15	1 1 1 1
#6	795	-.74	15.06	-70	4.2	-.28	16.1	2.94	77	24.87	1 1 1 1

*V: Velocity　　Θ: Pitch angle　　H: Altitude　　$\dot{H}$: Altitude rate

λ: LOS angle　　$\dot{\lambda}$: LOS rate　　AOT: Angle off　　R: Range

$\dot{R}$: Range rate　　SpEn: Specific Energy = $H + V^2/2g$

1: Is opponent in front?　　2: Am I behind opponent?

3: Can I see opponent?　　4: Can opponent not see me?

(1 indicates answer of yes, 0 no)

(Times were at 17, 22, 25, and 28 seconds)

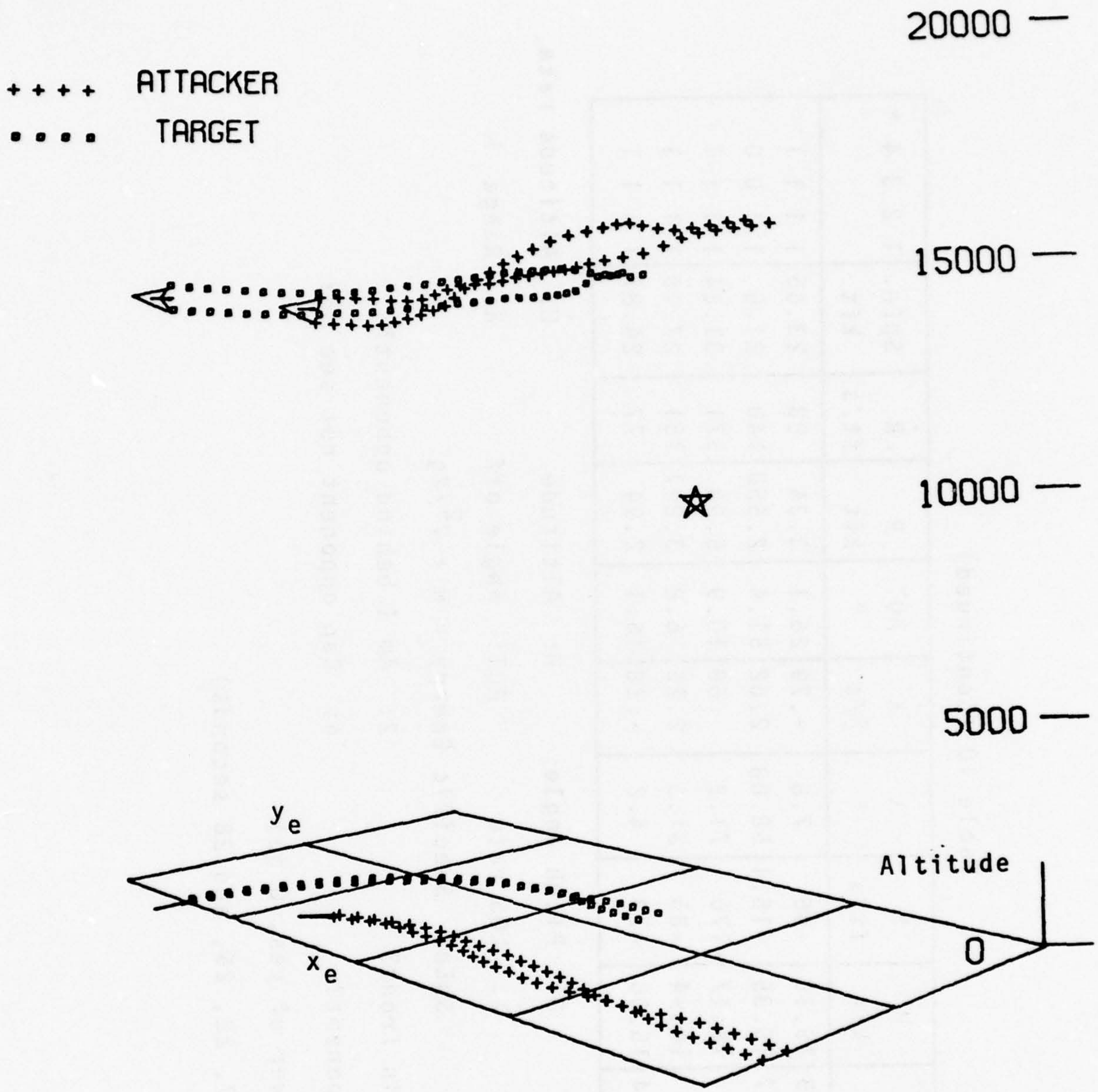
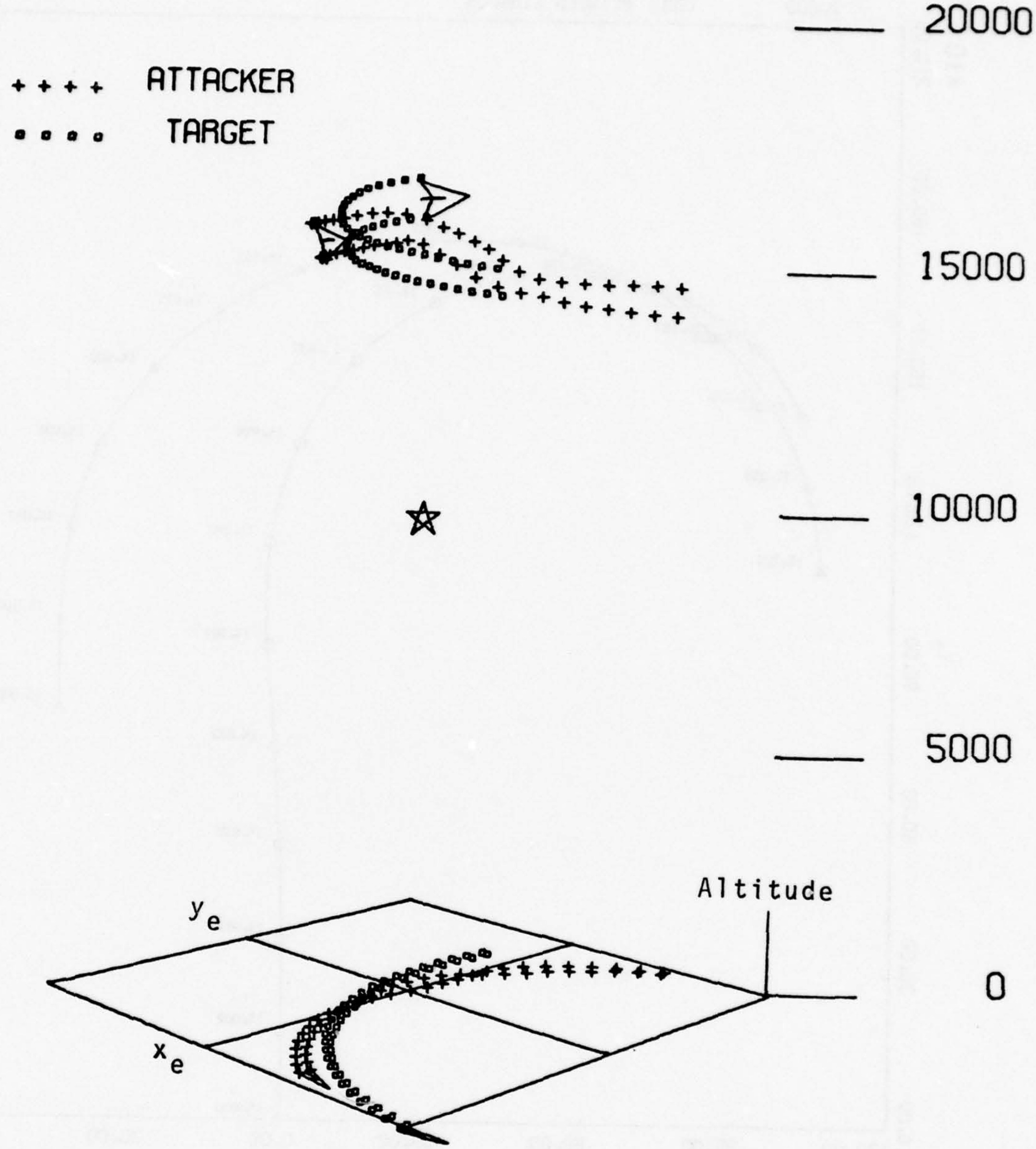


Figure 11. 3D plot and ground trace of AML executed low- and high-speed yo-yo 1 to 14.5 sec.

Range = 4053 feet at 14.5 Sec.
x_e and y_e are inertial coordinates



Range = 3121 feet at 28 sec.
x_e and y_e are inertial coordinates

Figure 11. (Continued) 15 to 28 sec.

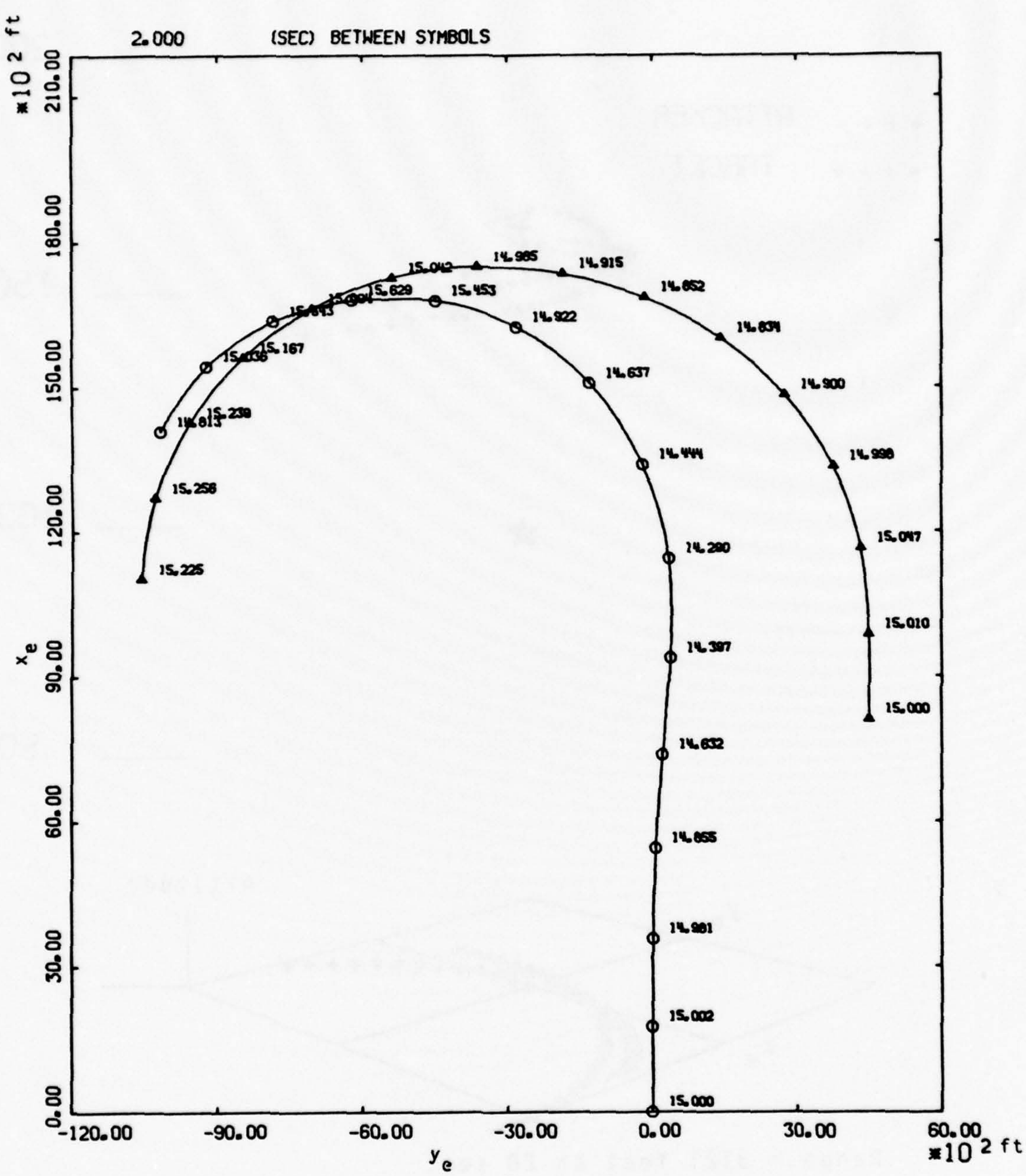


Figure 12. Ground trace of AML executed low- and high-speed yo-yo.

	Reference Yo-Yo	Case 7
Line-of-Sight Angle	7.82 deg.	8.34 deg.
Deviation Angle	5.80 deg.	18.64 deg.
Angle-off	26.09 deg.	7.01 deg.
Range	3,342 ft	2,703 ft
Range Rate	83 ft/sec	-77.5 ft/sec
Velocity	734 ft/sec	821 ft/sec
Altitude	15,178 ft	15,392 ft
Specific Energy	23,551 ft	25,880 ft

As was to be expected, the conditions at the termination of the yo-yo are more favorable for the AML flown case than they were for the pilot at the Luke AFB simulator. Starting the yo-yo with a higher speed, of course, provides an advantage to the AML program.

Reintroducing Questions and Weights

The preceding sections described how the AML program, when flying against a noninteractive target and when given appropriate command sequences in terms of load factors and bank angles, is capable of performing low-speed and high-speed yo-yo's superior to a human pilot. This is by no means a simple thing to accomplish, but it does not involve any application of the basic features of the AML program; that is, the defining of a set of importance-weighted questions, the consideration of trial maneuvers, scoring each maneuver by adding the sum of the weights attached to the questions satisfied by it, and then choosing the maneuver with the best score.

The technique to be applied requires that, for a given initial condition and a noninteractive target, a good referenced high-speed yo-yo is available.

As described before, the AML program will perform a maneuver selection in the following manner at various points in the yo-yo:

--Extrapolate the defender's position and altitude T_{pred} seconds ahead

--Select 3 to 6 trial maneuvers

--Predict own position and attitude for each trial maneuver

--Evaluate the outcomes of the trial maneuvers

--Execute the maneuver with the highest score

The crucial part of this process is the evaluation of the outcomes of the trial maneuvers. To apply the technique developed here for pilot training and evaluation, it is important that the different outcomes are evaluated with questions which have meaning to a pilot. These questions should concern variables either directly displayed to the pilot on the instrument panel (such as, heading, velocity, etc.) or relatively easily perceived by the pilot (such as, nose-tail separation and angle-off). Based on these criteria, the following list of eight questions to be asked to evaluate the situation at the end of the trial maneuver was derived:

1. Is my heading correct?
2. Is my altitude correct?
3. Is my climb (descent) rate correct?
4. Is my velocity correct?
5. Is my load factor correct?
6. Is my nose-tail separation correct?
7. Is range-rate correct?
8. Is angle-off correct?

The criteria for correct values are taken from the reference yo-yo's. These 8 explicit questions, which appear to be fairly independent, also contain answers to a number of implicit questions; such as, "Can I see my opponent?" etc. For each question, the program calculates the absolute value of the difference between the actual and reference value normalized to the maximum error in that question. It then multiplies this by a weight factor which will be assumed to remain constant during the entire maneuver. The total value for a given trial maneuver then will be:

$$V = 100 - \sum_{i=1}^{8} \left| \frac{\text{REFERENCE VALUE}_i - \text{ACTUAL VALUE}_i}{\text{MAX ERROR}_i} \right| \cdot W_i$$

Thus, if a trial maneuver would result in exactly the reference trajectory, its value would be 100; trial maneuvers which deviate in any of the eight criteria will have values less than 100.

Most trial maneuvers will deviate by different amounts in most of the 8 questions, and it is obvious that by changes in the weight factors, the rank ordering of the trial maneuvers will generally be changed (except for the unlikely case where one trial maneuver is worse than some other trial maneuver in <u>all</u> of the 8 criteria).

This is a drastic change and, hopefully, an improvement over the previous evaluation of the different trial maneuvers where all questions were binary in nature (yes or no). Many times, of the 6 trial maneuvers, only 2 different situations would occur; i.e., only 2 distinct sets of answers to the questions would occur. Now there should be a much better differentiation between the trial maneuvers.

SUMMARY

The objective of this contract is to develop a method which by observing pilot performance can determine the value or importance that he assigns to various performance criteria. In Phase I of this effort, the task was to develop techniques for using the AML program to compute this information from recorded performance data obtained by flying one AML program against another. The work reported here covers only Phase I. In Phase II the techniques developed here are to be used to compute information from actual pilot performance data.

One AML program, by observing the performance of another such program, produces a set of inequalities involving the question weights of the observed AML. A computer program with the inequalities as input gives a range of values for the weight of each question. The maximum values, the minimum values, or any linear combination which lies between them form a solution to the set of inequalities; i.e., if used as weights in an AML program, they would cause the program to perform the observed engagements exactly as the observed AML program. In general, for most questions, the bounds on the range of values were reasonably tight when the AML program was observed over several different engagements.

Discussions with fighter pilots confirmed that the relative geometry criteria of the AML program are not those used by the fighter pilot. In general, the pilots use standard air combat maneuvers dictated by the situation. Also, such pilot criteria as full use of airplane capabilities or energy management, for example, are of long-term evaluation and not compatible with the short-term decision scheme of the AML program. While the AML program does fly such maneuvers as a defensive turn or scissors, it does not fly more complex maneuvers such as, a high-speed yo-yo. For this reason, the AML program was modified so that it would execute a high-speed yo-yo against a noninteractive target. However, this involved introducing sequences of commands to the AML and did not use the basic AML logic. It is, therefore, necessary next to reintroduce questions and weights to fly the AML so that it performs a high-speed yo-yo. A list of such questions was given previously in this report.

In conclusion, one AML program by observing another AML program can, by using an LP program, obtain a set of weights equivalent to those used by the observed AML program;

i.e., one AML program can simulate another. However, it is not clear that the AML can find a set of weights which allow it to simulate the maneuvering of an actual pilot since differing maneuvers appear to require differing criteria.

REFERENCE

1. Burgin, George H., & Owens, A. J. An Adaptive Maneuvering Logic Computer Program for the Simulator of One-on-One Air-to-Air Combat, Volume II: Program Description, NASA Contractor Report NASA CR-2583, National Aeronautics and Space Administration, Washington, D.C., September 1975.

APPENDIX A

LINEAR PROGRAMMING

Linear Programming

The problem of determining the weights of the AML pilot by observing his actions led to a set of inequalities of the form $\sum_{i=1}^{15} a_i W_i \leq 0$ where each a_i is either -1, 0, or 1. In addition, each W_i must satisfy the inequalities $W_i \geq 1$ and $W_i \leq 5$. Geometrically, the set of values satisfying a given inequality is a half-space, and the set of values satisfying all of them is then the intersection of all these half-spaces. Finding a solution then reduces to finding a point in the intersection.

The situation here is typical of problems amenable to solution by the technique of linear programming. Such problems involve a set of parameters $W_i, \ldots, W_n$ with linear constraints of the form $\Sigma a_i W_i \leq b_i$ or $\Sigma a_i W_i \geq b_i$. In general, the constraints $W_i \geq 0$ are imposed. In our case this is redundant, since we have $W_i \geq 1$ as a constraint. Also, a linear function $f(w_1, \ldots, W_n) = \Sigma p_i W_i$, called the objective function, is given which has to be either minimized or maximized over the set of points (n-tuple) which satisfy the linear constraints. The set of points satisfying the constraints is termed the set of feasible solutions.

The technique of linear programming first proceeds to find a feasible solution. If no feasible solution exists, then the problem is not solvable. Once a feasible solution is found, one of several methods of finding the maximum or minimum solution is used. The most common is the Simplex method, and this is the one used in the study.

Once a feasible solution is found, an AML program using the solution as weights would react exactly over the test runs as the observed AML pilot. However, no information is given as to how close the feasibility solution is to the original set of weights. Since the scoring is done by adding the weights of the parameters with value 1, a natural function for an objective function is the sum of the weights; i.e., $\sum_{i=1}^{15} W_i$. The maximum and minimum solutions then give bounds on the possible values for each weight. Obviously, if all are equal, the solution is unique. In the various runs, while some bounds were tight, none yielded a unique solution.

Simplex Method. The boundary of the set of feasible solutions is in two dimensions a polygron, in three dimensions a polyhedron, and in higher dimensional spaces a simplex. Using the convexity property of the set of feasible solutions, it is straightforward to show that the maximum (or minimum) solution exists at one of the corner points (vertices) of the simplex. Note that the corner points are solutions to a system of simultaneous linear equations (a subset of the constraints of the problem considered as equations). The simplex method is a procedure for systematically examining the corner points until the optimum is found. The procedure uses operations involving pivot points similar to those used in solving simultaneous equations.

Initially, the constraints are converted into equalities by adding a new variable (a slack variable) to the less-than-or-equal-to constraint and by subtracting a new variable (a surplus variable) from the greater-than-or-equal-to constraints. So for example, if two constraints were:

(1) $5X_1 + 4X_2 \leq 200$

(2) $3X_1 + X_2 \geq 80$

they would be converted to:

(1') $5X_1 + 4X_2 + X_3 \quad 200$

(2') $3X_1 + X_2 - X_4 = 80$

For equations of type (1') the initial solution is $X_1 = X_2 = 0$ and $X_3 = 200$. This is not possible for type (2') since it would give $X_4 = -80$ which would violate the nonnegative requirement Several schemes exist for handling this. In the study, the "two-phase" method was used. This requires the addition of a second variable X_5 (an artificial variable) to (2') which then becomes

(2") $3X_1 + X_2 - X_4 + X_5 = 80$

so that a first solution is $X_1 = X_2 = X_4 = 0$ and $X_5 = 80$. In Phase 1, a dummy objective function involving the artificial variables is introduced and the optimum solution obtained for it. If all artificial variables are 0, then the solution is a feasible solution of the original problem. If not, no solution to the original problem exists.

To demonstrate the general technique, consider the problem:

Maximize: $X_0 = 8X_1 + 10X_2$

subject to:

$$5X_1 + 4X_2 \leq 200$$

$$3X_1 + 6X_2 \leq 180$$

$$4X_1 + 2.5X_2 \leq 108$$

The converted equations, including the objective function, are:

$$X_0 - 8X_1 - 10X_2 = 0$$

$$5X_1 + 4X_2 + X_3 = 200$$

$$3X_1 + 6X_2 + X_4 = 180$$

$$4X_1 + 2.5X_2 + X_5 = 108$$

with solution $X_1 = X_2 = 0$, $X_3 = 200$, $X_4 = 180$ and $X_5 = 108$ and value of objective function 0. The decision rule is to choose the variable with largest negative coefficient in the objective function, in this case, X_2, then in the constraints, choose the equations such that the ratio of the constant to the coefficient of X_2 is minimum. This would be the third equation (second constraint equation) so the element $6X_2$ is chosen as pivot element. Using $6X_2$ as pivot element eliminate the X_2 from the other equation. It is also conventional to divide the equation involving $6X_2$ by 6 so the resulting equations are:

$$X_0 - 3X_1 + 1.6667X_4 = 300$$

$$3X_1 + X_3 - .6667X_4 = 80$$

$$.5X_1 + X_2 \quad .1667X_4 = 30$$

$$2.75X_1 \quad .4167X_4 + X_5 = 33$$

and solution is $X_1 = X_4 = 0$, $X_2 = 30$, $X_3 = 80$, $X_5 = 33$ with value of objective function 300.

For the next step, the only variable with negative coefficient in the objective function is X_1 so it is chosen and the pivot element is $2.75X_1$ in the last equation. Using it as pivot element and eliminating X_1 from the other equations yields:

$$\begin{aligned}
X_0 \qquad\qquad\qquad &+ 1.2122X_4 + 1.0909X_5 = 336 \\
X_3 &- .2121X_4 - 1.0909X_5 = 44 \\
X_2 \qquad &+ .2424X_4 - .1818X_5 = 24 \\
X_1 \qquad\qquad & .1515X_4 + .3636X_5 = 12
\end{aligned}$$

with solution $X_1 = 12$, $X_2 = 24$, $X_3 = 44$, $X_4 = X_5 = 0$ and objective function value of 336. Since no negative coefficients exist in the objective function, this is the optimum solution.

In case there exist greater-than-or-equal-to constraints with artificial variables, say, X_{21}, X_{25}, and X_{30}, then in Phase 1 the objective function to be maximized is

$$X_0 = X_{21} + X_{25} + X_{30}$$

If it is a minimization problem then $X_0 = -X_{21} - X_{25} - X_{30}$ is minimized in Phase 1.

For minimization, in determining the pivot element the variable with the largest positive coefficient in the objective function is chosen; and if there are no positive coefficients, then the solution is optimal.

A listing of LP program follows.

```
      PROGRAM LINPRO
C
      COMMON /INEQLT/ INEQAL(100,2), INCNT, INEQFG
C
      DIMENSION CI(100),NXI(100),B(100),CJ(150),NXJ(150),A(100,150),
     1 Z(150),ZC(150),TITLE(20),CJ2(150),ITOC(100),SAVB(100),ISORS(100)
C
C        READ DATA
C
    1 READ(5,900) TITLE
      READ(5,902) M,NR,NORD,NO,KPRT,KSEND,KSENB,KODE,KROUN,IPFL
      DO 3 J = 1,150
    3 CJ2(J) = 0
      DO 4 I = 1,100
      DO 4 J = 1,150
    4 A(I,J) = 0
      IF(NO .EQ. 0)GO TO 11
C
C        GO TO ESTABLISH THE LIST OF ALL INEQUALITIES
C
      CALL INEQL1
   11 M = INCNT + 2 * NR
      IAF = 0
      KONE = 1
      KEND = 0
      NART = NR
      NSL = M - NR
      NSP = NR
      DO 3000 I = 1,NR
      ITOC(I) = 1
      B(I) = 5.
      SAVB(I) = B(I)
      ISORS(I) = 0
 3000 CONTINUE
      NR1 = NR + 1
      NR2 = 2 * NR
      DO 3010 I = NR1,NR2
      ITOC(I) = -1
      B(I) = 1.
      SAVB(I) = B(I)
      ISORS(I) = 0
 3010 CONTINUE
      NR3 = NR2 + 1
      DO 3020 I = NR3,M
      ITOC(I) = 1
      B(I) = 0.
      SAVB(I) = B(I)
      ISORS(I) = 0
 3020 CONTINUE
      IF(NORD - 1) 517,517,518
  517 N = NR + NSL + NSP + NART
      GO TO 519
  518 N = NR
  519 IF(NORD - 1) 520,530,545
  520 NC = NR + NSL + NSP
      DO 525 J = 1,NC
```

```
  525 NXJ(J) = J
      GO TO 550
  530 READ(5,902) (NXJ(J),J = 1,NR)
      JJ = NR + 1
      DO 540 I = 1,M
      IF(ITOC(I)) 535,540,535
  535 NXJ(JJ) = 800 + I
      JJ = JJ + 1
  540 CONTINUE
      GO TO 550
  545 READ(5,902) (NXJ(J), J = 1,NR)
      READ(5,902) (NXI(I), I = 1,M)
  550 DO 5 J = 1,NR
    5 CJ2(J) = 1.
    6 NMAX = M * N
      DO 3030 I = 1,NR
      A(I,I) = 1.
      A(I+NR,I) = 1.
 3030 CONTINUE
      DO 3055 I = NR3,M
      IAH = INEQAL(I-NR3+1,1)
      IBH = INEQAL(I-NR3+1,2)
      DO 3050 J = 1,15
      ITEMP = AND(IAH,2**(J-1))
      IF(ITEMP .EQ. 0)GO TO 3045
      A(I,J) = -1.
 3045 ITEMP = AND(IBH,2**(J-1))
      IF(ITEMP .EQ. 0)GO TO 3050
      A(I,J) = 1.
 3050 CONTINUE
 3055 CONTINUE
    8 IF(NORD - 1) 555,555,306
  555 J = NR + 1
      JJ = NR + NSL + NSP + 1
      DO 575 I = 1,M
      IF(ITOC(I)) 565,570,560
  560 A(I,J) = 1.0
      NXI(I) = NXJ(J)
      ISURS (I) = J
      J = J + 1
      GO TO 575
  565 A(I,J) = -1.0
      ISURS(I) = J
      J = J + 1
  570 A(I,JJ) = 1.0
      NXJ(JJ) = 900 + I
      NXI(I) = NXJ(JJ)
      JJ = JJ + 1
  575 CONTINUE
C
C          TEST FOR NECESSITY OF PHASE I
C
  306 IF(NXJ(N) - 900) 307,307,310
  307 IP2 = 0
      GO TO 330
  310 IP2 = 1
```

```
C
C         SETUP FOR PHASE I
C
      NART = 0
      DO 320 J = 1,N
      IF(NXJ(J) - 900) 315,315,316
  315 CJ(J) = 0.
      GO TO 320
  316 IF(KODE) 317,317,318
  317 CJ(J) = -1.0
      GO TO 319
  318 CJ(J) = 1.0
  319 NART = NART + 1
  320 CONTINUE
C
C         DETERMINE APPROPRIATE OBJECTIVE EQUATION
C
  330 IF(IP2) 331,331,10
  331 DO 335 J = 1,N
  335 CJ(J) = CJ2(J)
C
C         SETUP CI
C
   10 DO 15 I = 1,M
      DO 15 J = 1,N
      IF(NXI(I) - NXJ(J)) 15,14,15
   14 CI(I) = CJ(J)
   15 CONTINUE
      ITER = 0
C
C         COMPUTE Z AND ZC
C
   21 DO 25 J = 1,N
      Z(J) = 0.0
      DO 24 I = 1,M
   24 Z(J) = Z(J) + CI(I) * A(I,J)
   25 ZC(J) = CJ(J) -Z(J)
      OBJ = 0.0
      DO 28 I = 1,M
   28 OBJ = OBJ + CI(I) * B(I)
C
C         PRINT TABLEAU
C
   30 IF(KPRT) 101,101,31
   31 IF(IPFL .EQ. 0)GO TO 100
   32 IF(IP2)55,100,55
  100 IF(KONE) 55,55,101
  101 WRITE(6,916)
      WRITE(6,919)TITLE
      IF(IP2) 35,36,35
   35 WRITE(6,943) ITER
      GO TO 37
   36 WRITE(6,910) ITER
   37 N1 = 1
      N2 = 7
   43 IF(N2-N) 45,45,44
```

```
   44 N2 = N
   45 WRITE(6,911) (CJ(J),J = N1,N2)
      WRITE(6,912) (NXJ(J),J = N1,N2)
      DO 48 I = 1,M
   48 WRITE (6,913) CI(I),NXI(I),B(I),(A(I,J),J = N1,N2)
      WRITE (6,914) OBJ,(Z(J),J = N1,N2)
      WRITE (6,915) (ZC(J),J = N1,N2)
      IF(N2 - N) 52,55,55
   52 N1 = N1 + 7
      N2 = N2 + 7
      GO TO 43
   55 ITER = ITER + 1
      KONE = 0
      IF(KEND) 143,104,430
C
C          DETERMINE PIVOT COLUMN
C
  104 ZCM = ZC(1)
      JM = 1
      DO 109 J = 2,N
      IF(KODE) 105,105,105
  105 IF(ZC(J) - ZCM) 107,109,109
  106 IF(ZC(J) - ZCM) 109,109,107
  107 ZCM = ZC(J)
      JM = J
  109 CONTINUE
C
C          CHECK FOR OPTIMAL
C
      IF(KODE) 121,122,121
  121 IF(ZCM) 131,123,123
  122 IF(ZCM) 123,123,131
  123 IF(IP2) 400,429,400
C
C          CHECK FOR FEASIBILITY IN PHASE I
C
  400 IF(OBJ .LE. 1.0E-04 .AND. OBJ .GE. -1.0E-04) GO TO 405
      GO TO 427
  405 DO 410 I = 1,M
      IF(NXI(I) - 900) 410,410,415
  410 CONTINUE
      GO TO 429
C
C          DETERMINE PIVOT COLUMN TO ELIMINATE ARTIFICIAL VARIABLES
C             FROM BFS
C
  415 IM = I
      JM = 0
      XM = 1.0E40
      DO 423 J = 1,N
      IF(KROUN) 417,416,417
  416 IF(A(IM,J)) 418,423,423
  417 IF(A(IM,J) + 1.0E-04) 418,423,423
  418 IF(KODE) 420,419,420
  419 XX = ZC(J) / A(IM,J)
      GO TO 421
```

```
  420 XX = -1.0 * ZC(J) / A(IM,J)
  421 IF(XX - XM) 422,423,423
  422 XM = XX
      JM = J
  423 CONTINUE
      IF(JM) 146,427,146
  427 IAF = 1
C
C         INDICATE OPTIMALITY
C
  429 IF(KPRT) 430,430,124
  124 IF(ITER - 1) 430,430,168
  430 IF(IP2) 431,435,431
  431 IF(OBJ) 434,432,434
  432 IF(IAF) 434,433,434
  433 WRITE(6,917)
      IP2 = 0
      IF(IPFL .NE. 0)GO TO 436
      KONE = 1
  436 KEND = 0
      N = N-NART
      GO TO 330
  434 WRITE(6,941)
      GO TO 130
  435 WRITE(6,942)
  129 IF(KSENO) 1130,1130,200
 1130 IF(KSENB) 130,130,600
  130 IF(NO) 170,170,1
C
C         DETERMINE PIVOT ROW
C
  131 XM = 1.0E40
      IM = 0
      DO 139 I = 1,M
      IF(KROUN) 133,132,133
  132 IF(A(I,JM)) 139,139,135
  133 IF(A(I,JM) - 1.0E-04) 139,139,135
  135 XX = B(I) / A(I,JM)
      IF(XX - XM) 137,139,139
  137 XM = XX
      IM = I
  139 CONTINUE
      IF(IM) 141,141,146
  141 IF(KPRT) 143,143,142
  142 IF(ITER - 1) 143,143,167
  143 WRITE(6,918)
      IF(NO) 170,170,1
  146 IF(KPRT) 151,151,147
  147 KITER = ITER - 1
      IF(KITER) 1151,1151,148
  148 IF(KITER - 1) 149,149,150
  149 WRITE(6,916)
      WRITE(6,919) TITLE
  150 WRITE(6,920) KITER,OBJ
 1151 IF(A(IM,JM) - 1.0E-04) 251,251,151
  251 IF(A(IM,JM) + 1.0E-04) 151,252,252
```

```
  252 WRITE(6,940) A(IM,JM)
C
C       PERFORM PIVOT OPERATION
C
  151 XX = A(IM,JM)
      B(IM) = B(IM) / XX
      DO 154 J = 1,N
  154 A(IM,J) = A(IM,J) / XX
      DO 161 I = 1,M
      IF(I - IM) 157,161,157
  157 XX = A(I,JM)
      B(I) = B(I) - XX * B(IM)
      DO 160 J = 1,N
  160 A(I,J) = A(I,J) - XX * A(IM,J)
  161 CONTINUE
      CI(IM) = CJ(JM)
      NXI(IM) = NXJ(JM)
      GO TO 21
  167 KEND = -1
      GO TO 169
  168 KEND = 1
  169 ITER = ITER - 1
      KONE = 1
      GO TO 30
  170 CALL EXIT
C
C       SENSITIVITY ANALYSIS
C
  200 WRITE(6,916)
      WRITE(6,919) TITLE
      WRITE(6,930)
      DO 214 I = 1,M
      XMIN = -1.0E40
      XMAX =  1.0E40
      DO 207 J = 1,N
      IF(NXI(I) - NXJ(J)) 201,207,201
  201 IF(NXJ(J) - 900) 299,299,207
  299 IF(A(I,J) - 1.0E-04) 204,204,300
  300 IF(KODE) 205,202,205
  202 TLOW = ZC(J) / A(I,J)
      IF(TLOW - XMIN) 207,207,203
  203 XMIN = TLOW
      JSAVL = J
      GO TO 207
  204 IF(A(I,J) + 1.0E-04) 305,207,207
  305 IF(KODE) 202,205,202
  205 HIGH = ZC(J) / A(I,J)
      IF(HIGH - XMAX) 206,207,207
  206 XMAX = HIGH
      JSAVH = J
  207 CONTINUE
      TLL = CI(I) + XMIN
      UL =  CI(I) + XMAX
      IF(XMIN + 1.0E40) 209,209,208
  208 IF(XMAX - 1.0E40) 210,211,211
  209 IF(XMAX - 1.0E40) 212,213,213
```

```
  210 WRITE(6,931) NXI(I), CI(I),XMIN,NXJ(JSAVL),XMAX,NXJ(JSAVH),TLL,UL
      GO TO 214
  211 WRITE(6,932) NXI(I),CI(I),XMIN,NXJ(JSAVL),TLL
      GO TO 214
  212 WRITE(6,933) NXI(I),CI(I),XMAX,NXJ(JSAVH),UL
      GO TO 214
  213 WRITE(6,934) NXI(I), CI(I)
  214 CONTINUE
      IF(KSENB) 215,215,600
  215 IF(NO) 170,170,1
C
C         SENSITIVITY ANALYSIS OF B(I)
C
  600 WRITE(6,916)
      WRITE(6,919) TITLE
      WRITE(6,944)
      DO 675 I = 1,M
      IF(ITOC(I)) 601,670,601
  601 XMIN = -1.0E40
      XMAX =  1.0E40
      IF(NORD - 1) 605,605,610
  605 J = ISORS(I)
      GO TO 620
  610 DO 615 J = 1,N
      IF(NXJ(J) - ISORS(I)) 615,620,615
  615 CONTINUE
  620  DO 660 II = 1,M
      IF(A(II,J) - 1.0E-04) 640,640,625
  625 IF(ITOC(I)) 652,670,630
  630 TLOW = (-1.0 * B(II)) / A(II,J)
      GO TO 633
  632 TLOW = B(II) / A(II,J)
  633 IF(TLOW - XMIN) 660,660,635
  635 XMIN = TLOW
      JSAVL = II
      GO TO 660
  640 IF(A(II,J) + 1.0E-04) 645,660,660
  645 IF(ITOC(I)) 632,670,650
  650 HIGH = (-1.0 * B(II)) / A(II,J)
      GO TO 653
  652 HIGH = B(II) / A(II,J)
  653 IF(HIGH - XMAX) 655,660,660
  655 XMAX = HIGH
      JSAVH = II
  660 CONTINUE
      TLL = SAVB(I) + XMIN
      UL = SAVB(I) + XMAX
      IF(XMIN + 1.0E40) 662,662,661
  661 IF(XMAX - 1.0E40) 663,664,664
  662 IF(XMAX - 1.0E40) 665,666,666
  663 WRITE(6,945) I,SAVB(I),ITOC(I),NXJ(J),XMIN,NXI(JSAVL),XMAX,
     1 NXI(JSAVH),TLL,UL
      GO TO 675
  664 WRITE(6,946) I,SAVB(I),ITOC(I),NXJ(J),XMIN,NXI(JSAVL),TLL
      GO TO 675
  665 WRITE(6,947) I,SAVB(I),ITOC(I),NXJ(J),XMAX,NXI(JSAVH),UL
```

```
      GO TO 675
  666 WRITE(6,948) I,SAVB(I),ITOC(I),NXJ(J)
      GO TO 675
  670 WRITE(6,949) I,SAVB(I),ITOC(I)
  675 CONTINUE
      IF(NO) 170,170,1
C
  900 FORMAT(20A4)
  902 FORMAT(26I3)
  903 FORMAT(I3,F12.0)
  904 FORMAT(2I3,F14.0)
  905 FORMAT(2I3,F14.0,I3)
  910 FORMAT(1H ,≠ITERATION≠,I3(≠ OF PHASE II≠)
  911 FORMAT(1H ,34X,7F12.3)
  912 FORMAT(1H ,34X,7(7H     X(,I3,2H) ))
  913 FORMAT(1H ,F11.3, 4H  X(,I3,1H),3X,F12.3,7F12.3)
  914 FORMAT(1H0,18X,F16.3,7F12.3)
  915 FORMAT(1H ,34X,7F12.3///)
  916 FORMAT(1H1)
  917 FORMAT(1H0,≠OPTIMAL SOLUTION FOUND TO PHASE I≠)
  918 FORMAT(1H ,18HUNBOUNDED SOLUTION,/1H1)
  919 FORMAT(1H0,20X,20A4)
  920 FORMAT(1H ,≠ITERATION≠,I3,3X,≠OBJECTIVE = ≠,F16.3)
  930 FORMAT(1H0,T52,≠SENSITIVITY REPORT≠,/1H0,T47,≠LIMITING≠,T79,
     1 ≠LIMITING≠,/1H ,T15,≠ORIGINAL≠,T32,≠MAXIMUM≠,T47,≠VARIABLE≠,T64,
     2 ≠MAXIMUM≠,T79,≠VARIABLE≠,T97,≠LOWER≠,T112,≠UPPER≠,/1H ,
     3 ≠VARIABLE≠,T14,≠COEFFICIENT≠,T32,≠DECREASE≠,T45,≠OF  DECREASE≠,
     4 T64,≠INCREASE≠,T77,≠OF  INCREASE≠,T97,≠LIMIT≠,T112,≠LIMIT≠/)
  931 FORMAT(1H ,≠ X(≠,I3,≠)≠,T13,F12.3,T29,F12.3,T49,I4,T61,F12.3,T81,
     1 I4,T93,F12,3,T108,F12.3)
  932 FORMAT(1H ,≠ X(≠,I3,≠)≠,T13,F12.3,T29,F12.3,T49,I4,T62,
     1 ≠+ INFINITY≠,T81,≠----≠,T93,F12.3,T109,≠+ INFINITY≠)
  933 FORMAT(1H ,≠ X(≠,I3,≠)≠,T13,F12.3,T30,≠- INFINITY ≠,T49,≠----≠,
     1 T61,F12.3,T81,I4,T94,≠- INFINITY≠,T108,F12.3)
  934 FORMAT(1H ,≠ X(≠,I3,≠)≠,T13,F12.3,T30,≠- INFINITY≠,T49,≠----≠,
     1 T62,≠+ INFINITY≠,T81,≠----≠,T94,≠- INFINITY≠,T109,≠+ INFINITY≠)
  940 FORMAT(1H ,53X,≠PIVOT ELEMENT FOR NEXT ITERATION = ≠,E14.7)
  941 FORMAT(1H0,≠NO FEASIBLE SOLUTION≠/1H1)
  942 FORMAT(1H0,≠OPTIMAL SOLUTION FOUND TO PHASE II≠)
  943 FORMAT(1H ,≠ITERATION≠,I3,≠ OF PHASE I≠)
  944 FORMAT(1H0,T43,≠SENSITIVITY REPORT ON B(I) VALUES≠,/1H0,T58,
     1 ≠LIMITING≠,T83,≠_IMITING≠,/1H ,T9,≠ORIGINAL≠,T24,≠TYPE OF≠,T35,
     2 ≠CHANGE≠,T46,≠MAXIMUM≠,T58,≠VARIABLE≠,T71,≠MAXIMUM≠,T83,
     3 ≠VARIABLE≠,T98,≠_OWER≠,T112,≠UPPER≠,/1H ,≠ I≠,T11,≠B(I)≠,T22,
     4 ≠CONSTRAINT≠,T35,≠VECTOR≠,T46,≠DECREASE≠,T58,≠OF  DEC.≠,T71,
     5 ≠INCREASE≠,T83,≠OF  INC.≠,T98,≠LIMIT≠,T112,≠LIMIT≠/)
  945 FORMAT(1H ,I3,T7,F12.3,T25,I4,T36,I4,T44,F12.3,T60,I4,T69,F12.3,
     1 T85,I4,T94,F12.3,T108,F12.3)
  946 FORMAT(1H ,I3,T7,F12.3,T25,I4,T36,I4,T44,F12.3,T60,I4,T70,
     1 ≠+ INFINITY≠,T85,≠----≠,T94,F12.3,T109,≠+ INFINITY≠)
  947 FORMAT(1H ,I3,T7,F12.3,T25,I4,T36,I4,T45,≠- INFINITY≠,T60,
     1 ≠----≠,T69,F12.3,T85,I4,T95,≠- INFINITY≠,T108,F12.3)
  948 FORMAT(1H ,I3,T7,F12.3,T25,I4,T36,I4,T45,≠- INFINITY≠,T60,
     1 ≠----≠,T70,≠+ INFINITY≠,T85,≠----≠,T95,≠- INFINITY≠,T109,
     2 ≠+ INFINITY≠)
  949 FORMAT(1H ,I3,T7,F12.3,T25,I4,T35,≠EQUALITY CONSTRAINT--IF WANT AN
     1ALYSIS, CONVERT TO TWO INEQUALITIES≠)
      END
```

```
      SUBROUTINE INEQ.1
C
      COMMON /INEQLT/ INEQAL(100,2), INCNT, INEQFG
C
      DATA ISEVN/7777773/
      DATA ICARD/0/
C
      INCNT = 0
    5 READ 10,IAHAT,IBHAT
   10 FORMAT(O6,3X,O6)
      IF(IAHAT .EQ. ISEVN)GO TO 110
      ICARD = ICARD + 1
      IF(INCNT .EQ. 0)GO TO 70
C
C         COMPARE NEW *AHAT* WITH OTHER *AHAT* NUMBERS IN LIST *INEQAL*
C
      DO 50 K = 1,INCNT
      IAXOR = XOR(IAHAT,INEQAL(K,1))
      IAC = AND(IAXOR,INEQAL(K,1))
      IACTRL = AND(IAXOR,IAHAT)
      IF((IAC * IACTRL) .NE. 0)GO TO 50
      IBXOR = XOR(IBHAT,INEQAL(K,2))
      IBC = AND(IBXOR,INEQAL(K,2))
      IBCTRL = AND(IBXOR,IBHAT)
      IF(IBCTRL .EQ. 0 .AND. IAC .EQ. 0)GO TO 5
      IF(IBC .EQ. 0 .AND. IACTRL .EQ. 0)GO TO 80
   50 CONTINUE
C
C         PUT NEW EXPRESSION (*AHAT* AND *BHAT*) IN LIST *INEQAL*
C
   70 INCNT = INCNT + 1
      K = INCNT
   80 INEQAL(K,1) = IAHAT
      INEQAL(K,2) = IBHAT
      GO TO 5
  110 CONTINUE
      PRINT 130
  130 FORMAT(1H1)
      PRINT 135,ICARD
  135 FORMAT(1H ,I5)
      DO 150 I = 1,INCNT
      PRINT 140, INEQAL(I,1), INEQAL(I,2)
  140 FORMAT(1H ,O6,3X,O6)
  150 CONTINUE
      PRINT 130
      RETURN
      END
```

APPENDIX B

DATA TRANSFER PROGRAM

Data from the SAAC simulator at Luke Air Force Base were written on a 9-track magnetic tape reel. Since the AML program is on the CDC-3600 at the University of California at San Diego (UCSD) which will accept only 7-track magnetic tape, it was necessary to transfer the data from the 9-track tape to a 7-track tape. Fortunately, the Burroughs 6700 at UCSD handles both types of tapes and has a program to transfer data from one tape type to the other. Unfortunately, it was found that the standard program writes out the 7-track tape with even parity while the CDC-3600 accepts only odd parity. The addition of the proper control card to the transfer program corrected this problem.

A second problem encountered is that the data from the Luke Air Force Base Sigma-5 computer were in 32-bit floating point binary while the 3600 has a 48-bit word. When the data were read into the 3600, 3 32-bit Sigma 5 words were packed into 2 48-bit 3600 words and had to be unpacked into 3 36-bit words, right adjusted. This was readily done.

The remaining task was to convert each word into 48-bit floating point binary in the 3600. The exponent and mantissa were masked out and the exponent right adjusted. The sign bit was checked and reserved. Since the internal representation is 64 plus the actual exponent, 64 had to be subtracted from the exponent and then 16 raised to the result. The mantissa was initially designated as integer then floated and divided by 2^{24} to obtain the decimal representation; this was multiplied by 16 raised to the actual exponent power to get the 48-bit floating point binary representation. The sign of the number was determined by the reserved sign bit. The first few records were printed and compared with a data printout obtained at Luke Air Force Base. While the positive numbers were correct, the negative numbers had much too large absolute values. A check revealed that the exponent was essentially the complement of the one expected. So, for negative numbers the exponent was first complemented before being used as the desired exponent. While this made the negative numbers of the correct order of magnitude, they still did not agree with the Luke printout. A further check showed that the mantissa was also the complement of the expected one. Complementing the mantissa before the other computations gave correct results.

Briefly, the final program proceeds as follows (Let IN be the input word and OUT the output word):

```
ISIGN = RSHIFT(IN, 31)
IF(ISIGN.NE.0) IN = NOT (IN)
MANT = AND (IN, 2^24 - 1)
```

```
IEXP = RSHIFT (IN, 24)
IEXP = IEXP - 64
FMANT = MANT
OUT = (FMANT/2**24)*(16xxIEXP)
IF (ISIGN.NE.0) OUT = -OUT
```

☆U.S. GOVERNMENT PRINTING OFFICE: 1979- 671-056/58